An Artificial Intelligence Approach in Predicting Water Saturation in Carbonate Reservoirs

By

Wael Ateeq Al-Harbi

DECEMBER 2014

An Artificial Intelligence Approach in Predicting Water Saturation in Carbonate Reservoirs

BY

Waeil Ateeq Al-Harbi

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

PETROLEUM ENGINEERING

DECEMBER 2014

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **WAEIL ATEEQ AL-HARBI** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN PETROLEUM ENGINEERING.**

Thesis Committee

Dr. Abdulazeez Abdulraheem (Thesis Advisor)

Prof. Gabor Korvin (Member)

Dr. Enamul Hossain (Member)

Dr. Abdullah S. Sultan
(Department Chairman)

Dr. Salam A. Zummo
(Dean of Graduate Studies)

17/3/15
Date

Acknowledgments

Praise is due to Allah the creator of the universe and his prophet Mohammad peace be upon him, his family and all his companions.

Acknowledgment is owed to King Fahad University of Petroleum & Minerals for assisting this study, to Dr. Abdulazeez Abdulraheem for his continuous support, advises and for being my major advisor. I also would like to extend my thanks and appreciation to Dr. Enamul Hossain and Dr. Korvin Gabor for being in my thesis committee. I also appreciate and would like to thank Dr. Fatai Anifowose and Dr. Abdullah Al-Sultan the chairman of petroleum engineering department at KFUPM for their advices and encouragement.

Dedication

To my parents and my wife

خلاصة الرساله

اسم الطالب: وائل عتيق الحربي

عنوان الدراسة: استخدام الذكاء الاصطناعي في التنبؤ بتشبع الماء في المكامن الكربونيه

حقل التخصص: هندسة البترول

تاريخ الدرجة العلمية: نوفمبر 2014

تحديد حساب تشبع الماء يعتبر بتروفيزيائيا الأكثر تحديا. أيضا، تحديد تشبع الماء مهم جدا لأنه مطلوب كمدخل لحساب كميات النفط والغاز. هناك عدة طرق لحساب تشبع المياه على سبيل المثال، وذلك باستخدام المعادلة التجريبية مثل معادلة ارتشي، وكذلك باستخدام ضغط الشعريات في المختبر أو باستخدام العيانات لتحديد تشبع المياه.ويعتبر قياس تشبع المياه .باستخدام العيانات إذا ما تم تناوله بشكل صحيح متفوقة على غيرها من التقنيات فوق منطقة انتقالية النفط أو الغاز

نظام المسام في المكامن الكربونيه معقد جدا نظرا لوجود نظام مسامي ثانوي متغير بدرجة كبيرة. هذا جعل اكتساب وتحليل وتوصيف مجموعة البيانات البتروفيزيائية المكامن الكربونيه هو التحدي الاكبر

هناك العديد من تقنيات الذكاء الاصطناعي ولكن نحن في هذه الدراسه سوف نركز على التقنيات التي اظهرت نتائج ممتازه في صناعة النفط والغاز . جميع النماذج المستخدمه سوف تخضع الى عمليات تطوير لتحديد افضل نموذج. بيانات ابار وعيانات مكامن من بئرين كربونيين سوف تستخدم في هذه الدراسه. عمليات حسابيه تحليليه سوف تجرى على جميع النماذج المستخدمه للمقارنه. النموذج الحاصل على اعلى معدل ارتباط سوف يرشح للاستخام في صناعة النفط والغاز

Thesis Abstract

Student Name: Waeil Ateeq Al-Harbi

Title of Study: An Artificial Intelligence Approach in Predicting Water Saturation

Major Field: Petroleum Engineering

Date of Degree: November 2014

Determination of water saturation is one of the most challenging petrophysical calculations. Water saturation is very important because it is required as an input to calculate hydrocarbon volumes in a reservoir. There are several approaches to calculate water saturation, for example, using empirical equation such as Archie's equation, using laboratory capillary pressure or using core plug directly. Water saturation measured using core data if properly handled and preserved is superior to the other techniques.

Carbonate rocks have a very complex pore system because of the co-existence of interparticle porosity and highly variable secondary system of dissolution voids. This make the acquisition and analysis of petrophysical data and the characterization of carbonate rocks a big challenge.

In this study we have used four AI techniques to develop a model to predict water saturation in carbonate oil reservoir using well logs and core data. We are focusing on those AI techniques that have shown excellent results for similar problems in the oil and gas industry. All developed models are optimized to determine the best architecture based on the parametric analysis and input data sensitivity. The well log data and the core data obtained from two carbonate wells is used as the input for developing the model. Different statistical analyses are carried out on the

developed models for comparison purposes. The model with the maximum correlation coefficient and the minimum error is recommended for the practical applications in the industry.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Background

Every reservoir study needs petrophysics to lay down the foundation for better decision making. Petrophysics main focus interest is the analysis of the subsurface data measurements. Furthermore, petrophysics includes the investigation of well logs data that are coming from LWD (Logging While Drilling) wireline, formation testing and fluid sampling, special and conventional core data and mud logging. There are two ways to get petrophysical data: either through open hole or cased hole. In order to evaluate any oil or gas reservoir one need certain petrophysical properties. These are porosity, permeability, hydrocarbon saturation and reservoir intervals. This information is available in well logs data, core date and pressure data.

Information technology is well known to be the fast growing industry that is always at the head of innovation and advances. These innovative and advances concepts that come from the IT industry add a big tangible values which improve our life standards and living conditions. No other industry plays such a role where its impact can be seen and felt in almost every day of our lives. The oil and gas industry just like many other industries needs to capitalize on the available IT technologies.

One example of leveraging on the advancements of the IT industry is adapting the artificial intelligence techniques to help us improve our understanding of the subsurface geology. The application of the artificial intelligence techniques is wide spread (Helle et al 2002). It includes

planning, foreign language recognition and machine learning. An enormous value added can be achieved if our industry could capitalize on the machine learning science. Machine learning is very strong discipline which is related of finding non linearity relationships, optimization and statistics.

1.2 Statement of the Problem

The subject of petrophysics has a viewpoint of indirectness, in that, often, it is more challenging in many ways, financially, handling and expertise availability, to quantify an essential petrophysical reservoir property in nature. In addition, it has been estimated that conventional core can cost up to 10% of the drilling cost where sponge coring, type of coring for fluids saturation, can cost additional 10% of the drilling cost. Consequently, it is obvious that direct measuring of underground fluids and determining water saturation values from sponge cores is extremely costly and it requires extensive monitoring and continues of preparation and handling. At the same time, accurate water saturation data a measurement is considered one important challenge that the oil and gas industry is facing every day. Therefore, it is necessary to measure some other properties that are less challenging and related to the required property, in this case water saturation. It can be concluded that some of petrophysics background needed some procedures that link the possible petrophysical properties to the reservoirs properties.

One major challenge that the Petrophysicists who is specializes in carbonates can predict the petrophysical properties of carbonate reservoirs such as porosity and permeability. Even with all the excellent advancements in the oil and gas industry, we still lack a good understanding of the

relationship between porosity and permeability in carbonate reservoirs unlike the sandstone where this relationship is relatively understood. This is due to the fact that there are two pore systems exist together. The primary pore system includes interparticle porosity. Whereas the secondary pore system includes micro porosity. Another challenge that is related to the complex geology of the carbonate reservoirs is the challenge of optimizing the placement of newly drilled wells. All the above mentioned challenges increase the complexity of the carbonate reservoirs in many ways.

The importance of the accurate water saturation calculation comes in the form of determining the accurate calculation of the oil available in the reservoir using well logs data. Knowing this is the case, it is apparent that petrophysics play a major role when it comes to put together a strategic plan to optimize the production from the hydrocarbon field. Calculating hydrocarbon volume requires several petrophysical properties including water saturation which plays major role when it comes to field optimization plan. Therefore, accurately quantifying water saturation will greatly contribute to this task. It is worth mentioning that the most used techniques in the oil and gas industry to achieve this goal can be narrowed down to three major techniques. They are namely empirical resistivity models, Mercury Injection Capillary Pressure (MICP) experiments using capillary pressure experiments measurements and measuring the water saturation straight from the sponge cores

1.3 Thesis Objective

The objective of this study is to develop a cost effective new Artificial Intelligence (AI) model for the prediction of the water saturation in carbonate oil reservoirs using well logs data and core analysis data.

1.4 Approach

In this thesis we have used four AI techniques to develop a model to predict water saturation in carbonate oil reservoir using Matlab software. There are many AI techniques; however, we are focusing on the techniques that have shown excellent results in the past in the oil and gas industry. All developed models were studied for the sake of getting the best model.

The input layer consists of variables involved in the training and testing processes. The output layer contains the estimated water saturation from the sponge core.

The wireline well logs data and the measure sponge core data analysis belonging to two carbonate wells have used as the input for developing the models. Different statistical analysis will be carried out on the developed models for comparison purposes. The model with the highest correlation coefficient and lowest error will be recommended for the industry future utilization.

CHAPTER 2

LITERATURE REVIEW

2.1 Water Saturation Measurements

There are two techniques to calculate water saturation. The first technique is to calculate water saturation directly from core data. The second technique which is calculating water saturation indirectly is involving empirical equations and capillary pressure. In many situations it was reported that the water saturation values using these different techniques were not the same (Richardson 1994). As a result of this uncertainty the overall original oil/gas in place can result also in different values.

It was recommended that these techniques should be carried out at the same time, if funds are available, and then uncertainty analysis should also be performed.

2. 2 Water Saturation from Empirical Models

Over the past years many researchers came up with different empirical models that claim best description of the relationship between well log data and water saturation. However, the majority of these empirical models where just an extension of the well-known Archie's equation. The components for this equation are:

$$Sw = \left(\frac{aR_w}{R_t \phi^m} \right)^{\frac{1}{n}}$$

Sw : Water saturation (V/V)

Rw : Formation connate water resistivity (OHMM)

φ : calculated porosity from the wireline well log data (V/V)

Rt : Resistivity measurement from well log data (OHMM)

n : Saturation exponent (Unit less)

m : Cementation exponent (Unit less)

a : Cementation constant (Unit less)

Since 1942 when Archie first introduced his equation, the majority of the industry has adopted this equation for water saturation calculations. This model becomes the basis for well log analysis. In most of the circumstances the petrophysical empirical equation are built around sandstone reservoirs. Not only that, it was reported several time that these sandstone reservoirs are called "Archie" reservoirs. The main requirements for the validity of Archie's equation can be summarized in Table 2-1:

Table 2-1 Main Archie's Equation Requirements

Archie Criteria	Non-Archie condition
Intergranular porosity	(Micro)fractures/fissures/vugs porosity
Homogenous	Heterogeneous
single mode porosity	Multiple mode porosity
Water wet	Mixed wettability

There are many situations that could prevent from applying Archie's equation correctly. This includes the shaliness, multimode porosity system and low resistivity or fresh water in the formation. Without considerations of the above situations could lead to lose of funds and effort

by underestimation water saturation. However, there are some ways to complement the shortcoming of applying Archie's equation on the carbonate reservoirs such as using nuclear magnetic resonance (NMR) or elemental tool. In some case core data will also be required. All of this effort is put together to minimize the uncertainty in the calculation of water saturation from this empirical model. Other commonly used mathematical models are the Simandoux and dual water models which are based on sandstone exclusively and are derivative of Archie's equation. There are limitations of these empirical methods. One of the limitations of these models is selecting the right model that best describes the relationship of the water saturation, formation resistivity and connate water resistivity. One needs to keep in mind that these empirical models are approximations of the real nature of the reservoir pore system. Furthermore, the input parameters such as the saturation exponent 'n', cementation exponent 'm' and Rw are in many cases considered constant where as they are variables in reality (Bust 2009).

2.3 Water Saturation from Laboratory Capillary Pressure

This method of calculating water saturation capitalizes on the capillary equilibrium which has formed during the Geologic Period. There are three methods to reproduce this equilibrium in the laboratory namely MICP, porous plate and centrifuge. This method is independent of well logs which makes it more appreciated due to its relative high accuracy and cost effectiveness.

There are several shortcomings of this method. First of all, one could question the procedure that was followed in the laboratory to perform this experiments and whether enough time was allocated to reach equilibrium or not. Also, one could question whether the conversion from surface condition to reservoir condition was carried out correctly with the correct interfacial tension values

Moreover, it has been reported that it is very challenging especially in carbonate reservoir the distinguish between the free water level and the observed oil water contact (OWC). Finally this shortcoming is also true for all small scale measurement such as core data is whether enough data were obtained and whether this data is representing the targeted reservoir.

(Richardson 1994).

2.4 Water Saturation from Core Data

Before the advancement of the oil and gas industry many approaches to analyze some issues were primitive. For example, there was a need to have a closer look to the lithology that was embedding the hydrocarbon. Because of that need some geologists and drilling engineers invented the coring techniques in the Netherlands in 1908. Also, in the past many geoscientists and petroleum engineers used to crack the core that they acquired. Then they look for any sign of hydrocarbon existence by smelling and tasting these small pieces.

When core analysis first introduced many geoscientists and petroleum engineers were skeptical about this new approach of analysis. In recent years, core acquisition, processing and analysis has its own specialty in the industry. Most of the reservoir study projects are using core data to integrate it in their study along with wireline well logs data. A special type of such coring methods is sponge coring. The main objective of the sponge coring is to minimize and eliminate the loss and evaporation of all potential subsurface fluids.

Direct measurements of water saturation are performed by using oil based mud sponge core data and applying Dean-Stark water-volume methodology. There are some limitations to this technique. For example, the well must be drilled with oil base mud. Moreover, it has been

reported that there are big uncertainty of this data that were obtained close by the oil water contact (Richardson 1994).

Also, in addition the high cost of this method there is another potential risk that could jeopardize the whole project. This risk is the handling and the preserving of the core and whether extra caution was took during the entire procedure. Mishandling the cores could result in false data reading.

2.5 Artificial Intelligence

The implementation of the AI models to solve complicated problems has been highly respected in many industries including the oil and gas industry. Some of these complicated problems were challenging and at some point were slowing down improvements and advances in our industry. AI modeling techniques were utilized and seen the added value in different disciplines within the oil and gas industry. For example, AI was developed to predict several subsurface geological and petrophysical properties such as permeability and porosity prediction. Also, AI models were built to identify seismic pattern, improve oil and gas production and analyze drill bit problems. (Bazzaz 2007; Anifowose 2013; Bello 2014)

2.5.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a nonlinear algorithmic, non-digital, and intensely parallel data processing. ANN is following similar approach as the human nerve system in the way how to process data and information. Warren McCulloch who was a neurophysiologist

along with Walter Pitts who was a mathematician were the pioneer of discovering the ANN and that was in 1943.

ANN is capable of constructing a solid relationship between input data and output data. Regardless of the type of this relationship and whether it is nonlinear or linear relationship ANN will build an equation that describes this relationship. This is credited to its learning competence through the back propagation mechanism that ANN is implementing. The application of ANN is widespread. It can fill missing data, predict a physical property or recommend most optimize plan. Figure 2-1 shows the structure of a typical ANN.

$$y_i = f(\sum_k w_{ik} x_k + \mu)$$

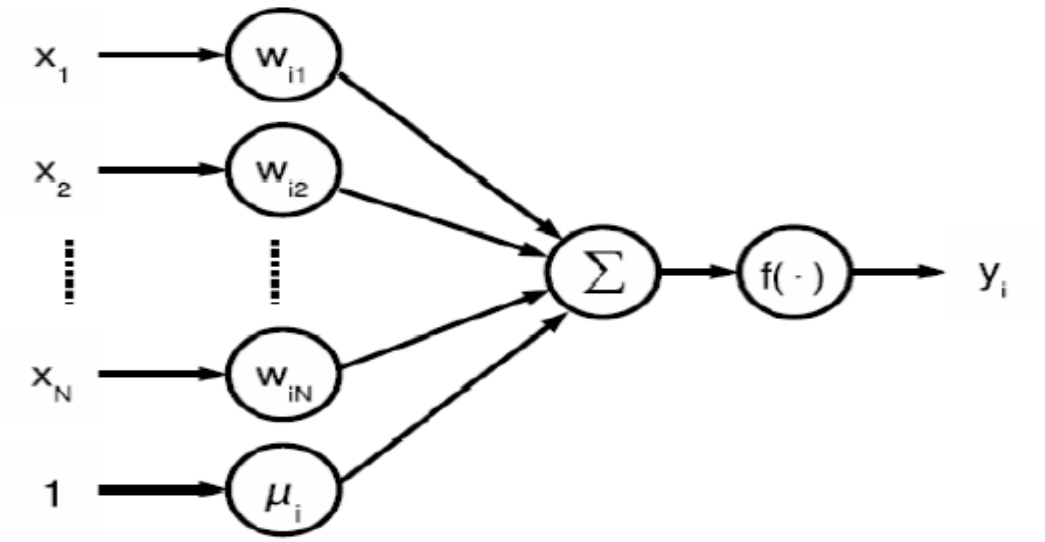


Figure 2-1 The structure of artificial neural network

2.5.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised artificial intelligence model with related kernel functions that investigate input data and identify patterns. SVM is well known for its excellent clustering and classification but it can also be used for prediction. This algorithm was presented by Vladimir N. Vapnik in 1963 for linear models and then was extended by Cortes and Vapnik in 1995 to nonlinear cases.

SVM looks among different classes and clusters of data points and then position them in the space to extend as possible that space between different classes. The closest input point to that touch the separation line is named support vector.

Figure 2-2 shows the schematic SVM model and how it is mapping the input data cluster in a higher space in dimensionality.

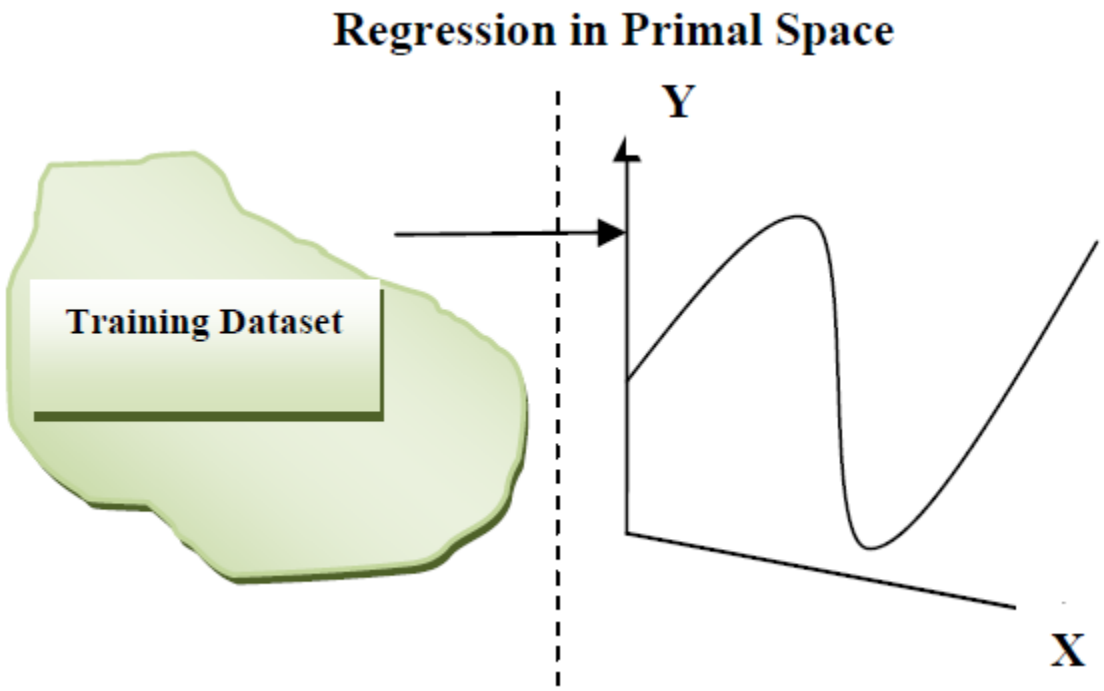


Figure 2-2 Schematic of SVM model

Moreover, SVM has the ability to address the non-linear problems as good as addressing the linear problems. In this case SVM uses what is called kernel that facilitates the hyperplanes mapping of the input data.

SVM is a popular alternatives to other AI techniques because of its ease of training, simple architecture and stability to converge. There is an excellent competence of SVM where it finds the best separation of the data which resulted in excellent clustering of the input data. At the same time SVM could be weak due to its need for a good kernel function.

2.5.3 Functional networks

By looking to the schematic of the functional network any one can see the great similarity between neural network and functional network. It is true that functional network is an extension of neural network but there are fundamental differences. For example, in neuron network there are weights associated to the hidden neurons which can be learnable where in functional network there is no weight however the functions in the hidden neurons that the functional network neuron is using is able to adjust and learn. Also, one of the distinguish advantage of the neuron network is the ability of the back propagation where in the case of functional network is no back propagation and it is actually only feed forward.

Functional network modeling uses MDL algorithm (minimum description length) which is capable of selecting the best sub-set from the input data. The main criteria that MDL algorithm is implementing to choose the sub-set input data is based on the best non-linear relationship between each input data and the output data. This could result of overall enhancement to the

model since weak related input data to the output data will be discarded before starting building the functional network model. At the same time functional network is using the least square statistical approach to optimize this relationship.

There are two types of functional network as figure 2-3 shows. These two types are called serial functional network and the other one is called one-layer functional network. The difference between the two types is the complexity of the algorithm that each one is implementing.

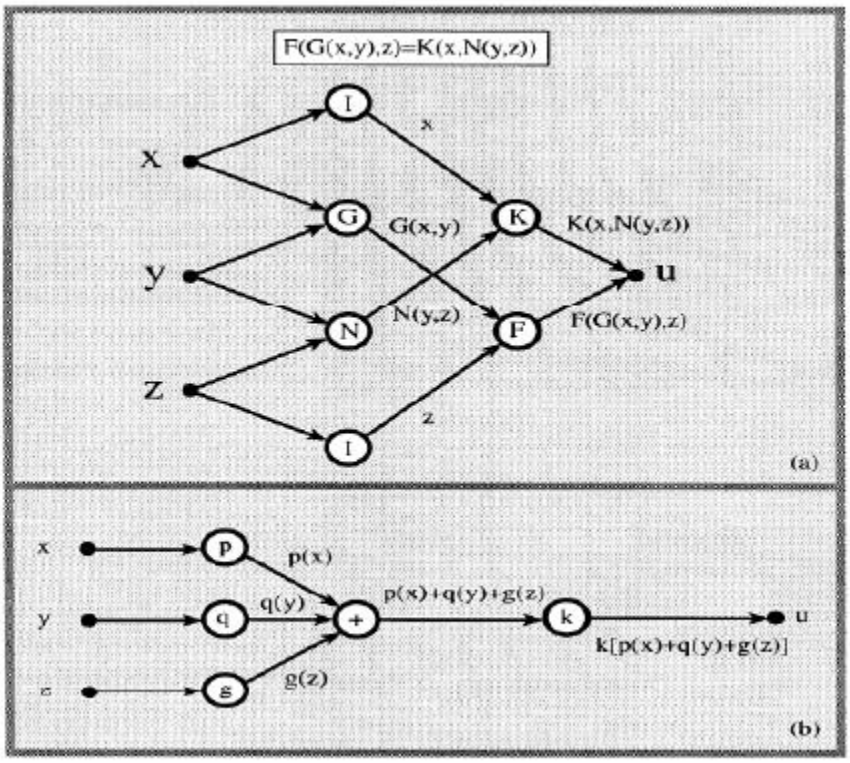


Figure 2-3 Schematic of both types of Functional Networks

2.5.4 Fuzzy Logic

The AI model fuzzy logic gained its name due to its mechanism of dealing with the data. It has some similarity with other AI model of the way it is describing the data. In other words, human being uses relative words to describe things or situations such as weak, soft dark and warm. The absolute meaning of these words is different from one person to another and hence the term "fuzzy". Fuzzy logic modeling uses this approach in the way it is describing reservoir properties as well as other natural properties. Its main goal is to establish a relationship between input and output data and to decrease the uncertainty. Not only that, fuzzy logic model will classify the input data into several clusters as well as assigning certain percentages of membership of the same input data into other clusters.

The concept of Fuzzy Logic evolved from the fuzzy set theory proposed by a mathematician of Iranian descent, Dr. Lotfi A. Zadeh in 1965. Dr. Zadeh is considered the father of Fuzzy Logic and its implementations in mathematics, computer sciences, system control and artificial intelligence. Unlike the crisp logic where the false and true is implemented, the fuzzy logic modeling utilizes different approach to tackle its modeling problems. It assigns the variables in subject certain values that enclosed between 0 and 1. These values are called the truth values.

The relation between a variable and its truth value can be described by a "membership" function that ranges between 0 and 1 and the value of this variable in the functions defines a "degree" of membership. The type-2 Fuzzy Logic System (T2FLS) is more advanced that the type-1 FLS where the membership grades are themselves fuzzy. Also, the other membership function has its own limit which extends between (0, 1). Figure 2-4 shows the structure of Type-2 FLS which provides a new degree of dealing with uncertainties.

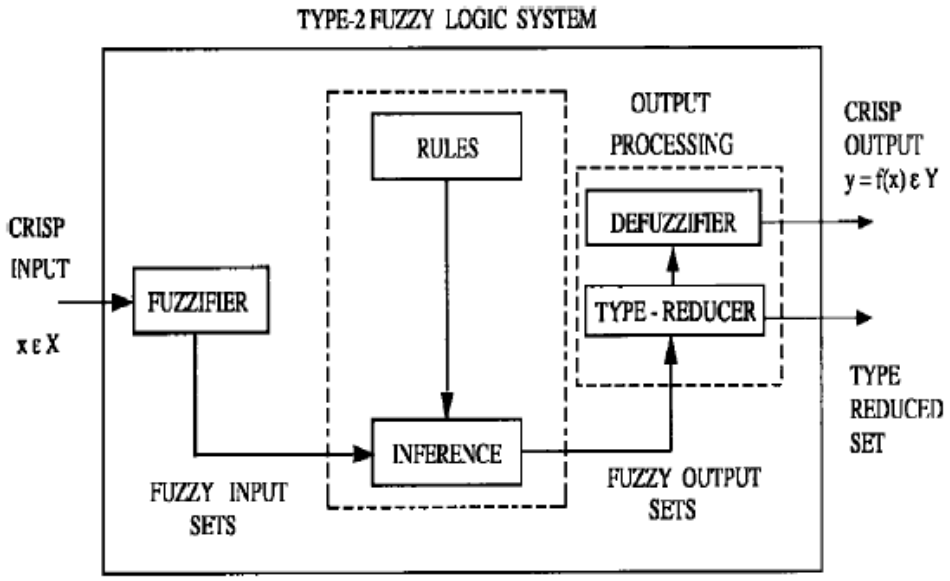


Figure 2-4 Structure of a Type-2 FLS

2.5.5 General Framework

Figure 2-5 shows the conventional workflow of the different AI models that was utilized in this study. Simply when using AI models and once the input data and the target data were identified then certain percentage of the data will be assigned as the training data and the remaining data will be assigned as the testing data.

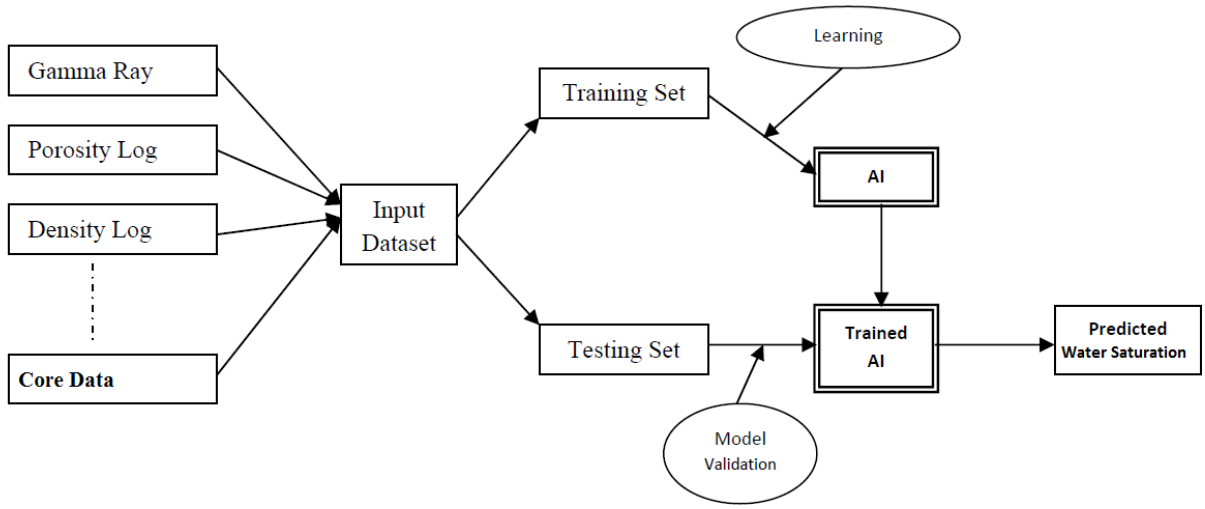


Figure 2-5 General Workflow of AI Systems

2.6 Artificial Intelligence in the Oil & Gas Industry

Many papers have addressed the utilization of different models of artificial intelligence in the oil and gas industry. AI can add great value added benefit to the professional of the oil and gas industry. Moreover, AI can play major role in solving conventional petroleum engineering problems where conventional approaches have difficulties and challenges. It was reported in the literature that the implementation of the artificial intelligence modeling techniques should be utilize when there is a strong justification. For example, using artificial intelligence modeling techniques to solve issues that is related on how one should treat his/her pet or his/her neighbors is not advisable.

One of the earliest implementation of the AI in the oil and gas industry was done by Helle and Bhatt in 2002. In this study they developed in artificial neural network model to predict the subsurface fluids (water, oil and gas). They used wireline well logs data. The result of the ANN model test against the actual data values was excellent. In 2007, Moustafa and Hamada developed an ANN model using well logs and core data for their target aim of interpreting and

evaluating the reservoir formation. The built neural network presented excellent porosity and water saturation similarity with the core data. In 2009, Al-Bulushi et al used two sandstone reservoirs data from the Middle East in his study to build an artificial neural network (ANN) models using only wireline open hole data to estimate water saturation in these reservoirs. After that, he applied the ANN model that he built and tested it against other sandstone reservoirs. The result of the test showed the strong prediction capabilities of the ANN model that he developed.

SVM has been successfully utilized in the gas and oil industry to forecast properties of the reservoir such as porosity and permeability. El-Sebakhy et al. (2007) used SVM technique to predict the most important PVT properties which are the pressure of the bubble point and the volume factor of the oil formation. Not only that, the developed SVM outperformed the common published empirical correlations. Anifowose and Abdulraheem (2010) have highlighted the strength of the SVM to work in a small dataset and the simplicity of training. Merits of SVM are its ability to converge on global optima and stable performance. Furthermore, Anifowose et al (2011) have showed the over-performance of SVM comparing to other AI techniques for predicting permeability and porosity in carbonate reservoirs.

Functional Networks have been implemented successfully in the oil & gas industry. In 2014 Bello and Asafa used Functional network modeling to forecast bottom hole flowing pressures and temperatures in vertical multiphase wells using 700 data points from multiple fields. The study concluded that the functional network models can predict flowing buttonhole pressure and temperature in production wells under wide range of operating conditions.

Type-2 fuzzy logic (T2FLS) is a new advancement of the fuzzy logic scheme. Anifowose and Abdulraheem (2010) were successful in using T2FLS to predict two gas and oil reservoirs

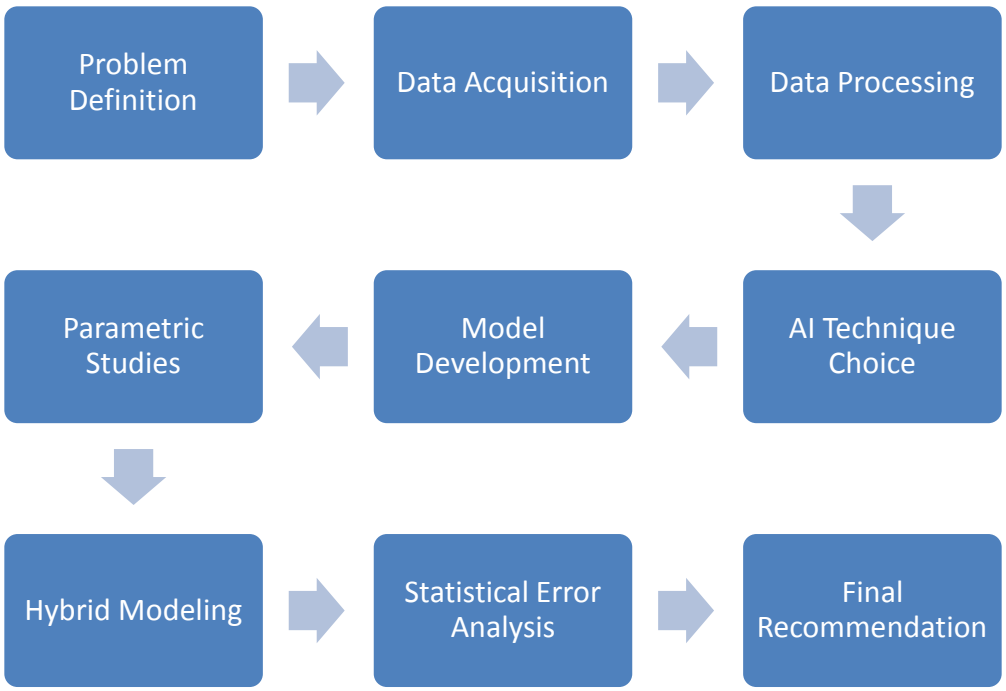
properties namely permeability and porosity. Anifowose and Abdulraheem (2013) used T2FLS to predict permeability by getting the most out of seismic and well log data to increase the capability of improving the prediction accuracy.

In this thesis, all AI techniques as mentioned above are needed to predict water saturation. In order to enhance the results and to extract maximum information from input data, hybrid modeling is also employed. Furthermore, to help AI models in improving the predictions, saturation outputs from Archie's equation are used as additional inputs.

CHAPTER 3

MODEL DEVELOPMENT

There are two items will be discussed in this chapter. First, all data associated items such as data acquisition, data preparation and data processing are introduced. After that, the developed AI that was used in this study will be discussed thoroughly. A proposed workflow has been implemented to streamline the models development.



3.1 Data Acquisition

A key step to ensure the success of any AI modeling is the data preparation and management. Setting us some key quality assurance and control can lead to the increase of the model prediction capabilities. The theme here is to build a mathematical model using artificial

intelligence approach to predict water saturation in carbonate reservoirs. Building an artificial intelligence model requires relatively large data since all artificial intelligence techniques are data driven. Well log data and core data are used as input data into these models. These are considered as confidential data in oil and gas companies. Not only that, in many cases all the data which were found belonged to clastic reservoirs or "sandstone reservoirs" whereas the focus of this thesis is carbonate reservoirs. Further, water saturation from core data is very rare data type due to its high cost of measurements. Acquisition of these data was a major milestone toward execution of this thesis work. Thankfully, two carbonate well logs were acquired and used in this study with 198 data points from core analysis. The core data was received overburden corrected smoothed and depth matched resulted in good match with wireline porosity. As for the wireline data, it included sonic log, resistivity log, density log, PEF log, porosity log and gamma ray log.

3.2 Data Preparation and Processing

The goal of all artificial intelligence models is to come up with model that is general enough to be used in similar scenarios and situations. A small computer program was used utilizing a stratification function which helps generalize the model by imposing input data randomness. Another benefit of running the stratification function is to avoid any bias of the data in representing the model. As for the data division, the common practice was followed in this thesis of dividing the input data set into two sets. The training set was included of 70% of the training data and the other data which is the 30% was used for validating the model.

Toward ensuring data uniformity and consistency and removing any outliers additional steps were taken.

A color map was created for each input feature to point out any outliers' data or data spikes that would be possibly negatively affect the results. Figures 3-1 through 3-6 are plots of the input data versus depth that belong to the Well No. 1. Terms and notations given in the figures are explained below.

DEPTH : Depth in feet

DT : Compressional wave travel time

PHIE : Effective porosity

RT : Resistivity

RHOB : Bulk Density

GR : Gamma Ray

PEF : Photoelectric Factor

SW : Water saturation

Going through these figures one can be see that the data is in good shape and there is no apparent abnormality that could necessitate taking correction actions. The data is falling within its natural ranges which indicate good quality of the data.

Another set of color maps were created for each input feature to point out any outliers or data spikes that would be possibly negatively affect the results for Well No. 2. Figures 3-7 through 3-12 are plots of the input data versus depth. Looking at these figures one can be see that there is no apparent abnormality in the data that could necessitate taking correction actions. Each parameter is within its natural range.

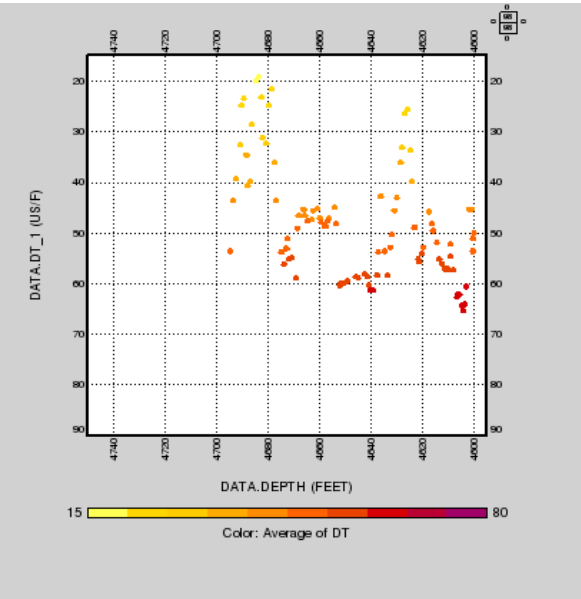


Figure 3-1 DT color coded with its limit for well #1

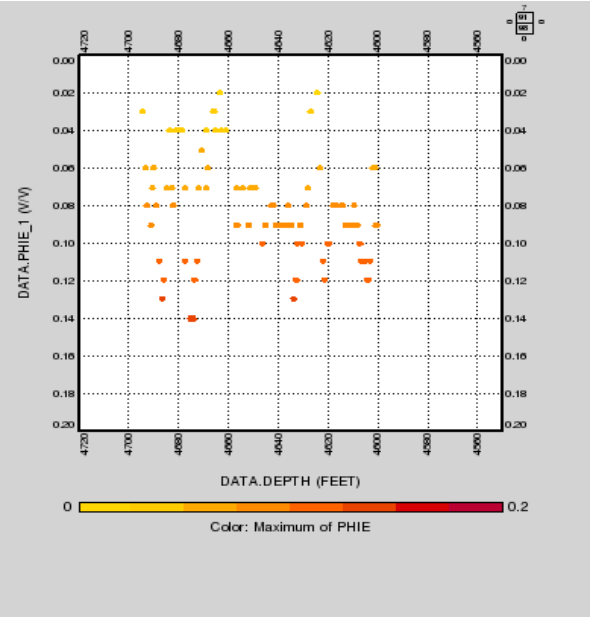


Figure 3-2 PHIE color coded with its limit for well #1

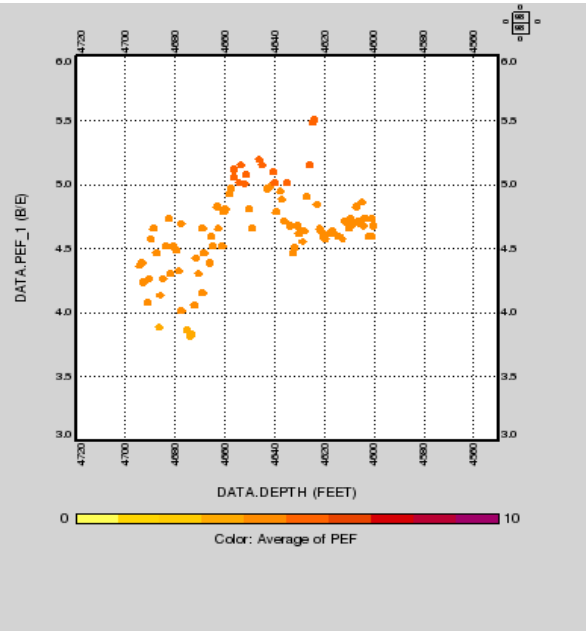


Figure3-3 PEF color coded with its limit for well #1

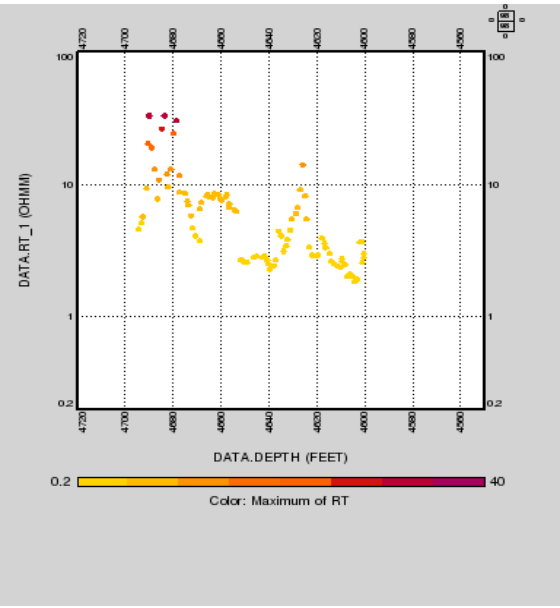


Figure 3-4 RT color coded with its limit for well #1

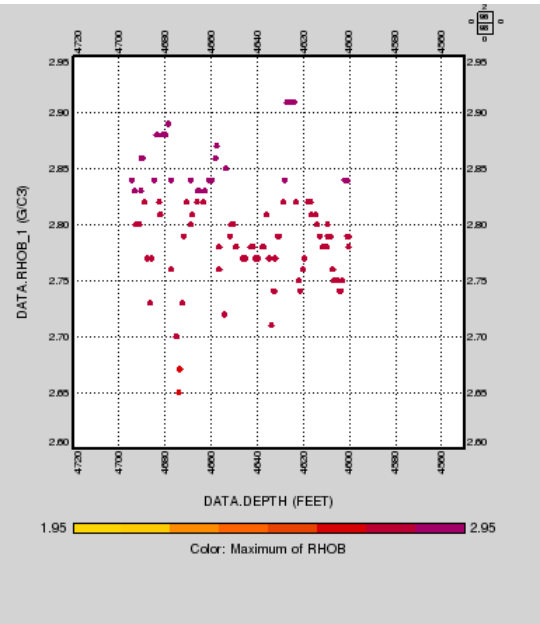


Figure 3-5 RHOB color coded with its limit for well #1

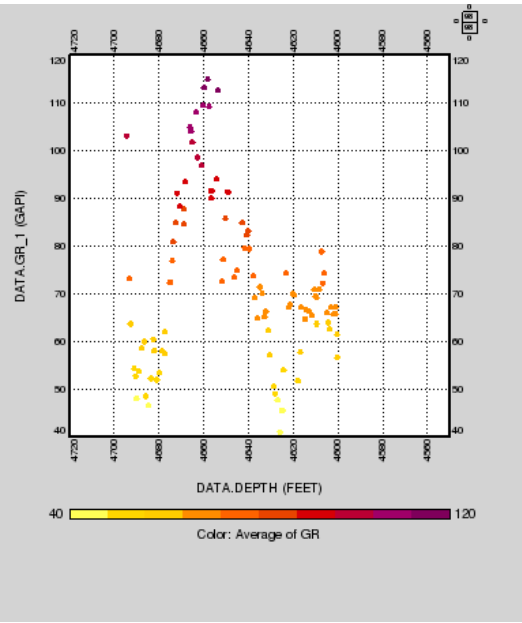


Figure 3-6 GR color coded with its limit for well #1

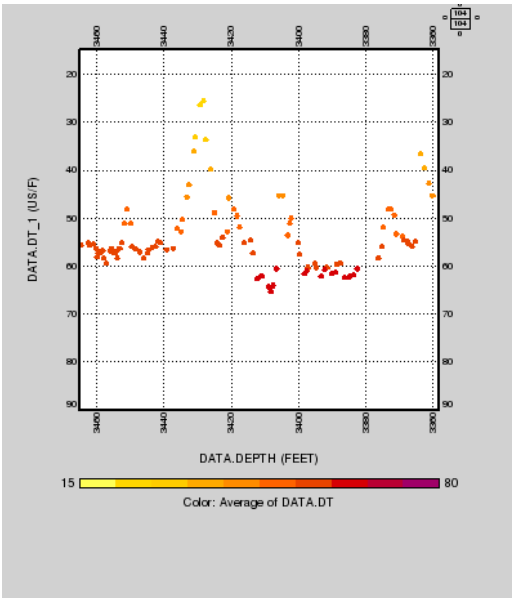


Figure 3-7 DT color coded with its limit for well #2

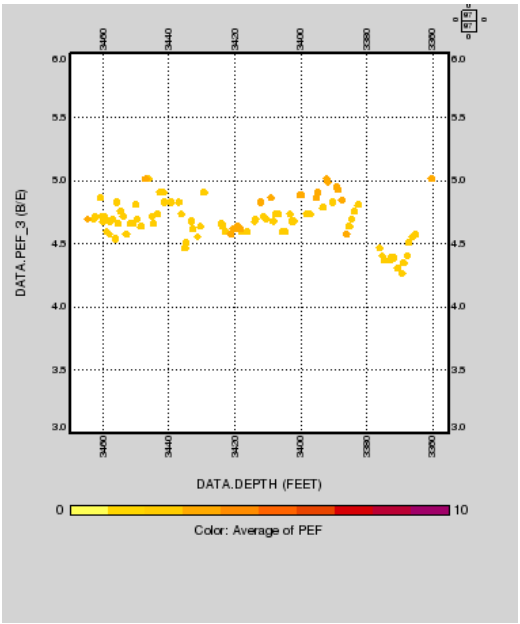


Figure 3-8 PEF color coded with its limit for well #2

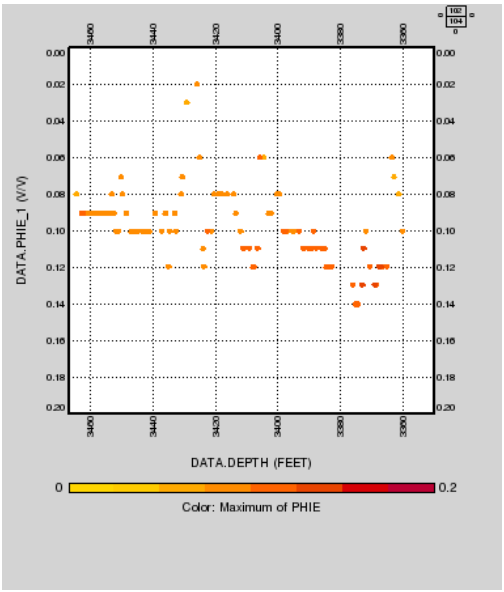


Figure 3-9 PHIE color coded with its limit for well #2

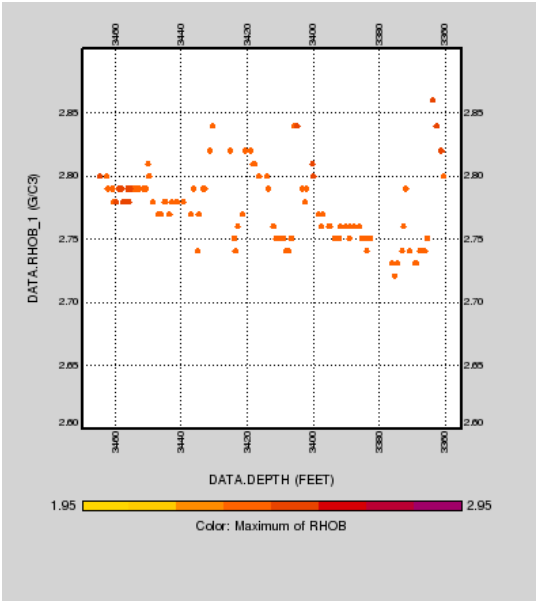


Figure 3-10 RHOB color coded with its limit for well #2

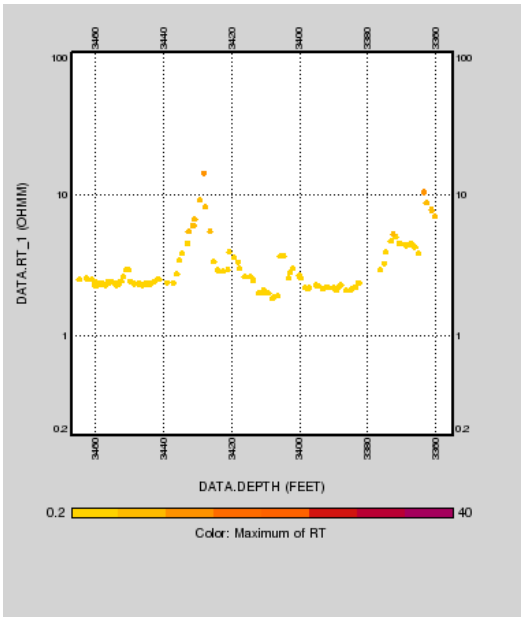


Figure 3-11 RT color coded with its limit for well #2

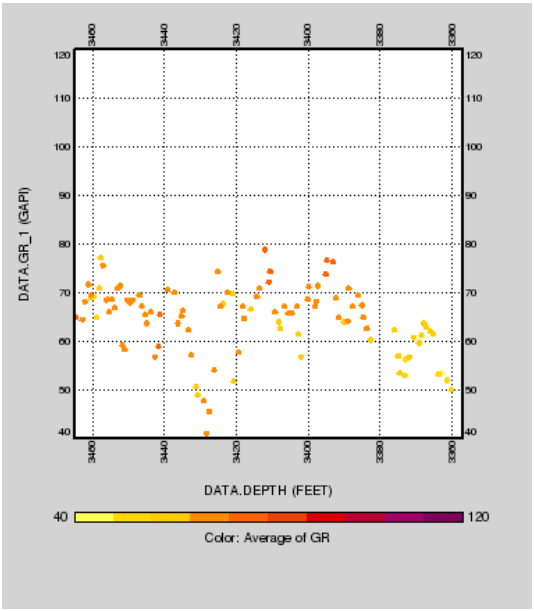


Figure3-12 GR color coded with its limit for well #2

3.3 Measurements of Error

Several statistical error approaches were implemented to appraise the predicted log water saturation measurement (Sw) using the newly developed AI techniques by bringing the estimated results from each technique along the real measurements of water saturation from core analysis data in the same view. The statistical error parameters that were studied are: Correlation Coefficient (CC), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE).The equations for these parameters are given below

1. Correlation Coefficient

The failure or success in minimize the standard deviation can be symbolizes by the correlation coefficient . It has a value ranging between 0 and 1. It is given by

$$R = \frac{\sum_{i=1}^{n}\left(SW_m - \overline{SW_m}\right)\left(SW_p - \overline{SW_p}\right)}{\sqrt{\sum_{i=1}^{n}\left(SW_m - \overline{SW_m}\right)^2 \sum_{i=1}^{n}\left(SW_p - \overline{SW_p}\right)^2}}$$

Where SW_m is the measured water saturation, $\overline{SW_m}$ is the mean of the measured water saturation, $\overline{SW_p}$ is the mean of the predicted water saturation and SW_p is the predicted water saturation.

2. Root Mean Square Error:

To calculate the distribution nearby zero deviation the root mean square was utilized. The formula is showing below as:

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n} {E_i}^2}$$

where in the above equation *n* is the value of testing samples and E_i is the difference between the real values and the predicted values.

3. Maximum Absolute Percentage Error

This error measures the maximum relative deviation from among all data samples. It is defined as:

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{SW_m - SW_p}{SW_m}\right|$$

where SW_m is the measured water saturation value, SW_p is the predicted value of water saturation and n is the number of tests.

4. Mean absolute error

One useful statistical approach to test the prediction model accuracies is called the mean absolute error (MAE). The MAE can inform the research how far or how close is the predicted value from the original value. The MAE equation is below

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|SW_p - SW_m\right|$$

where SW_{p_i} is the estimated value and SW_m is the actual value.

3. 4 AI Model Development

In this thesis, MATLAB application software is the main tool that was used to develop the AI models. Many industries such as aviation and pharmaceutical use Matlab to model solve and visualize the different problems that they are facing in their everyday activities. Understanding the problem that one is about to tackle is very important in order to meet the intended objective of developing an AI models. In this study water saturation prediction in carbonate oil reservoir is the planned target. Not only that, we are planning to achieve better results that the conventional models.

Four Artificial intelligence techniques were utilized in this study to come up the best AI model for the problem. These AI techniques are Artificial Neuron Network (ANN), SVM, FN and Fuzzy logic. As a first run of the models, all available data were used. Graphical representation of the results provides a quick and adequate understanding of the model prediction performance. The measured and predicted water saturation values are plotted for all the training and testing samples to indicate the excellent fit between them and demonstrate the robustness of the developed AI models. To further analyze the results graphically, additional representations are generated. This includes measured and predicted water saturation versus depth crossplots.

Cross-plots provide graphical representations of the correlation quality between the actual and predicted water saturation values. In this study, a cross plot was created for all the measured values and the predicted values plotted on the other axis. A line that cross the (0, 0) point was plotted in these cross plot to graphically determine the relative agreement between the measured value and the predicted values.

Figure 3-13 shows the crossplot of measured and predicted water saturation values (Sw) in the testing phase using ANN. In these studies, 70% of the input data was used for training and 30%

of the dataset was used for testing and validation. It can be seen from the plot that there is a special trend in the prediction. When the measured water saturation is low the prediction was relatively good however when the measured water saturation is high (>0.7) the prediction model was under estimating the water saturation. Figure 3-14a and figure 3-14b show the measured and the predicted water saturation plotted against the depth. One can see that well#2 is relatively in a better agreement with the model than well#1

Figures 3-15 and 3-16 represent similar results for Type-2 Fuzzy Logic System (T2FLS). In figure 3-15 one can see that the prediction is quite bad at low saturation values; For middle values, the prediction is fine. However, for higher saturations, the model was over predicting. Looking at the measured water saturation and predicted water saturation plot versus depth in Figure 3.16a and figure 3.16b, one can see that the model was not in agreement at all with well#1 where it was relatively in agreement with the deeper part of well#2.

Figures 3-17 and 3-18 show the result for the functional network. Looking at these figures one can see that the result is much better than those for the previous model. However, the model is over predicting the water saturation. Finally Figures from 3-19 and 3-20 show the result for the Support vector Machine (SVM). One can see from these figures that the prediction is good at lower measured water saturation with slightly over prediction. At the end of this exercise, several charts are presented to compare different statistical errors among all the used AI techniques. These charts are shown in Figure 3-21 through Figure 3-24. In case of correlation coefficient, one can see that ANN has outperformed all the other AI techniques and this confirms report in the literature about the robustness of ANN. In the other statistical error measurements, such as the root mean square, mean absolute error, and the maximum absolute percentage error ANN was in the lead by having the least values for all these statistical measurements. Moreover,

one can see that the functional networks were performing nearly as good as ANN where T2FLS was the most underperforming model.

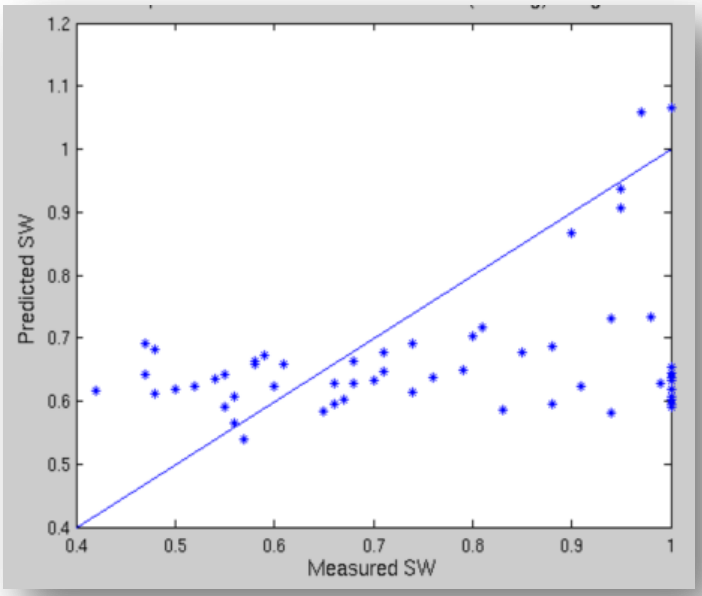


Figure 3-13 Crossplot of Measured and Predicted Sw (Testing) using ANN

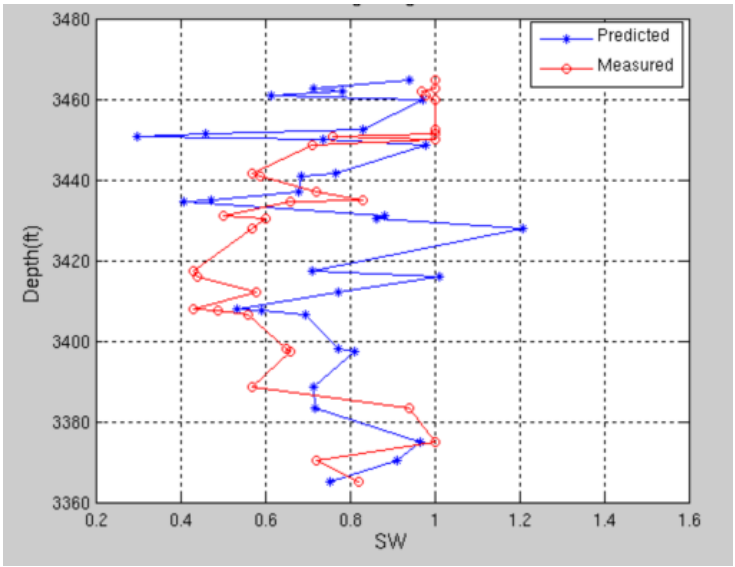


Figure 3-14a Depth vs Sw Testing using ANN Model for well#1

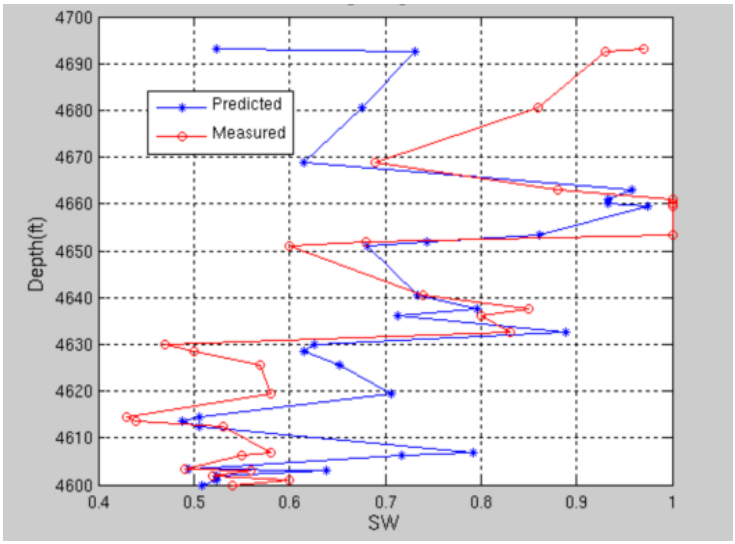


Figure 3-14b Depth vs Sw Testing using ANN Model for well#2

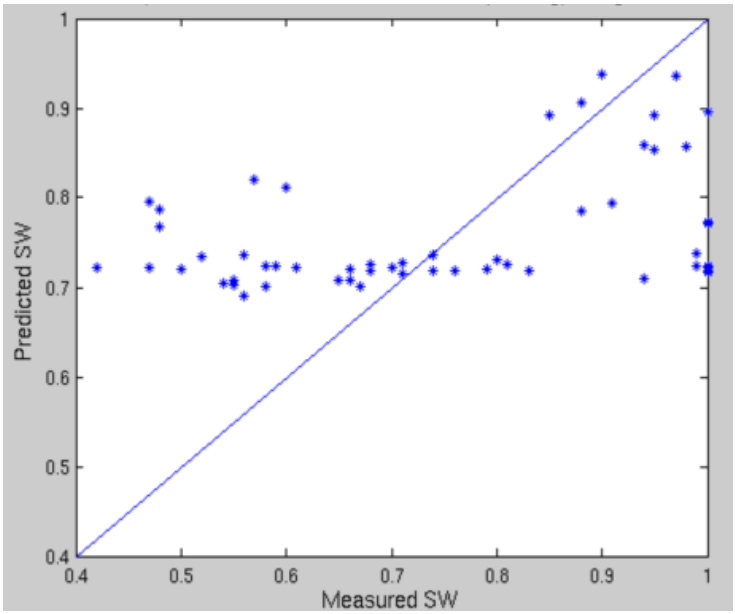


Figure 3-15 Crossplot of Measured and Predicted Sw (Testing) using T2FLS

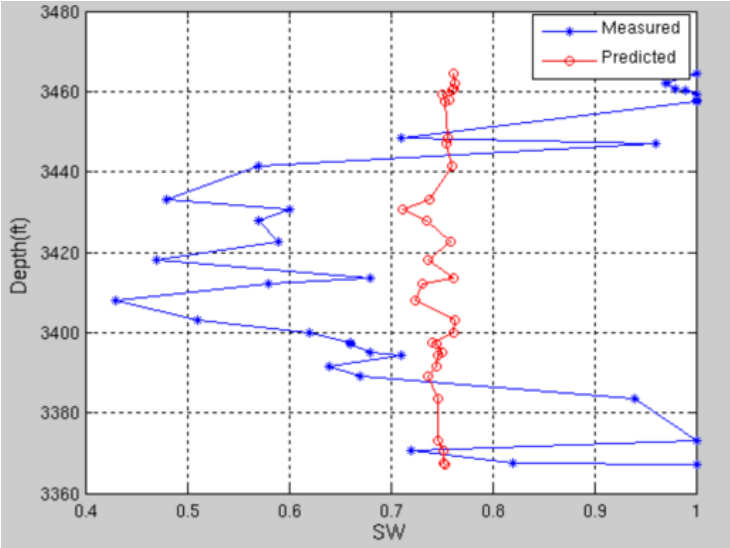


Figure 3-16a Depth vs Sw Testing using T2FLS Model well#1

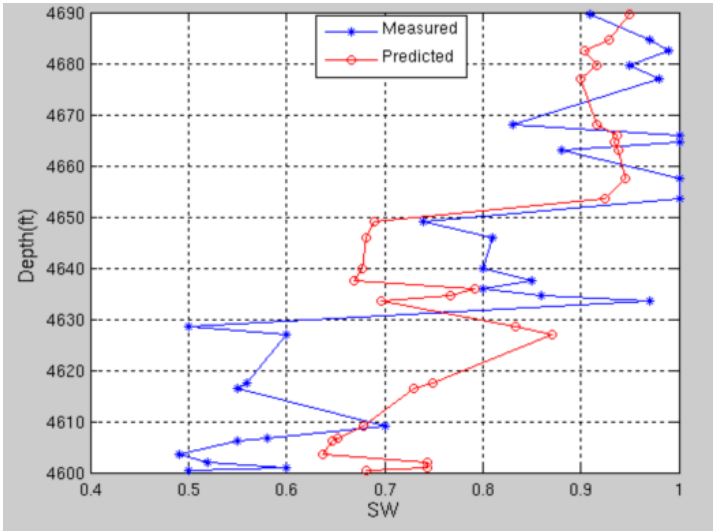


Figure 3-16b Depth vs Sw Testing using T2FLS Model well#2

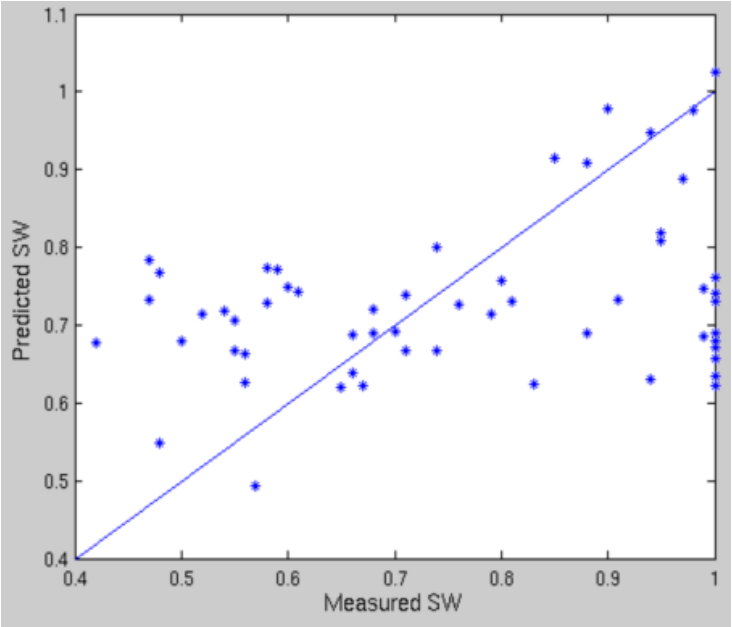


Figure 3-17 Crossplot of Measured and Predicted Sw (Testing) using FN

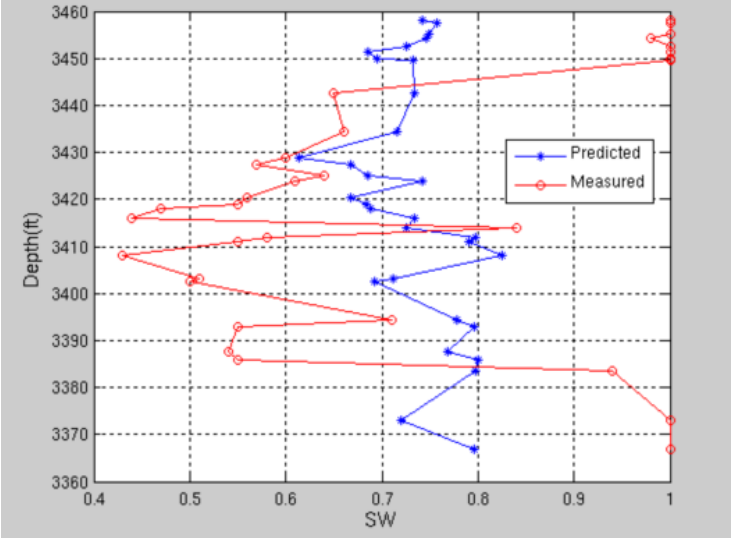


Figure 3-18a Depth vs Sw Testing using FN Model for well#1

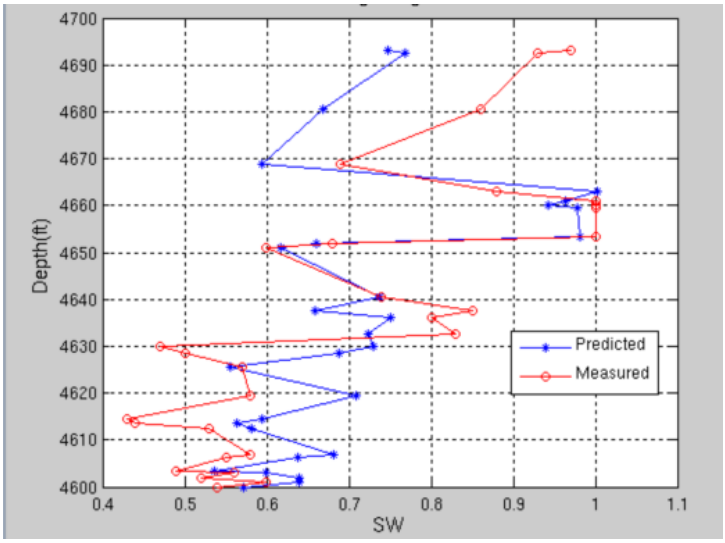


Figure 3-18b Depth vs Sw Testing using FN Model for well#2

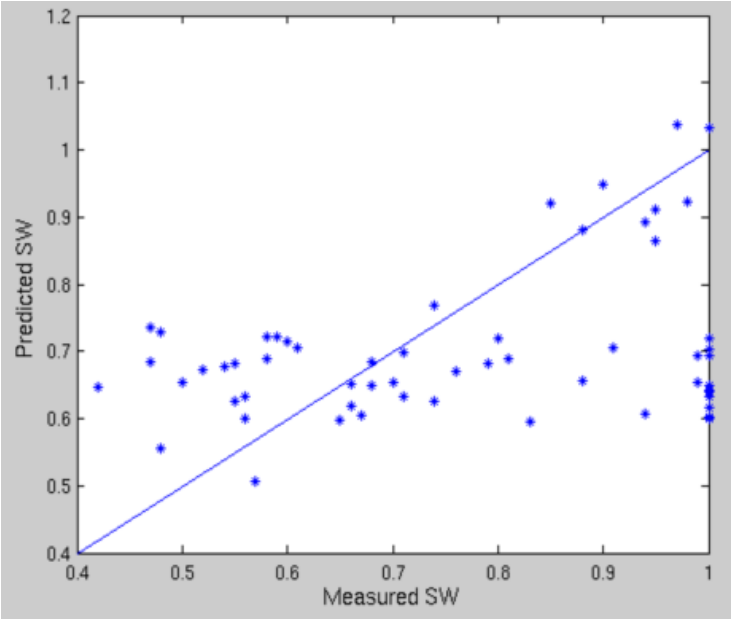


Figure 3-19 Crossplot of Measured and Predicted Sw (Testing) using SVM

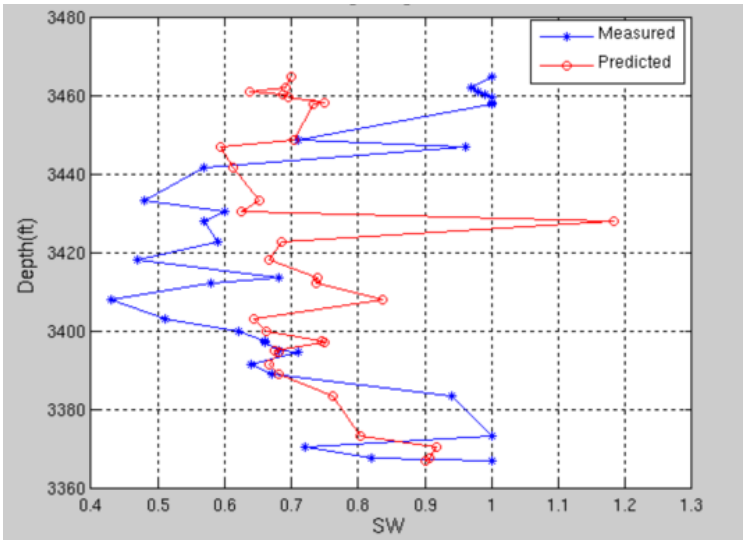


Figure 3-20a Depth vs Sw Testing using SVM Model for well#1

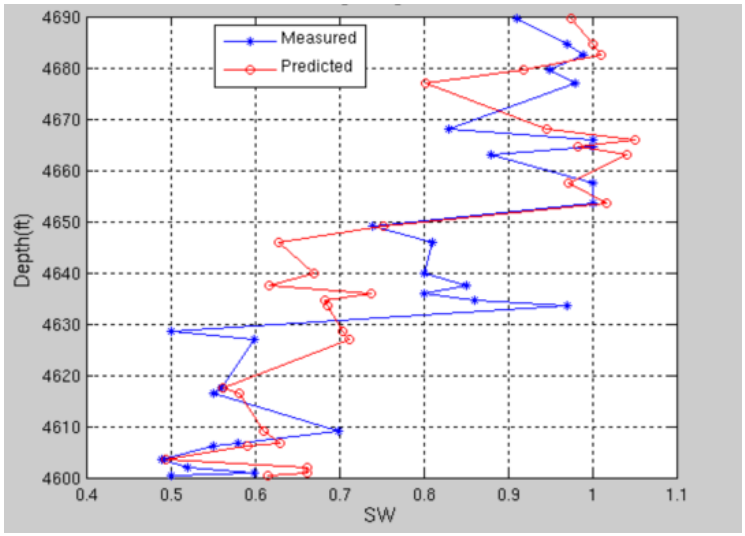


Figure 3-20b Depth vs Sw Testing using SVM Model for well#2

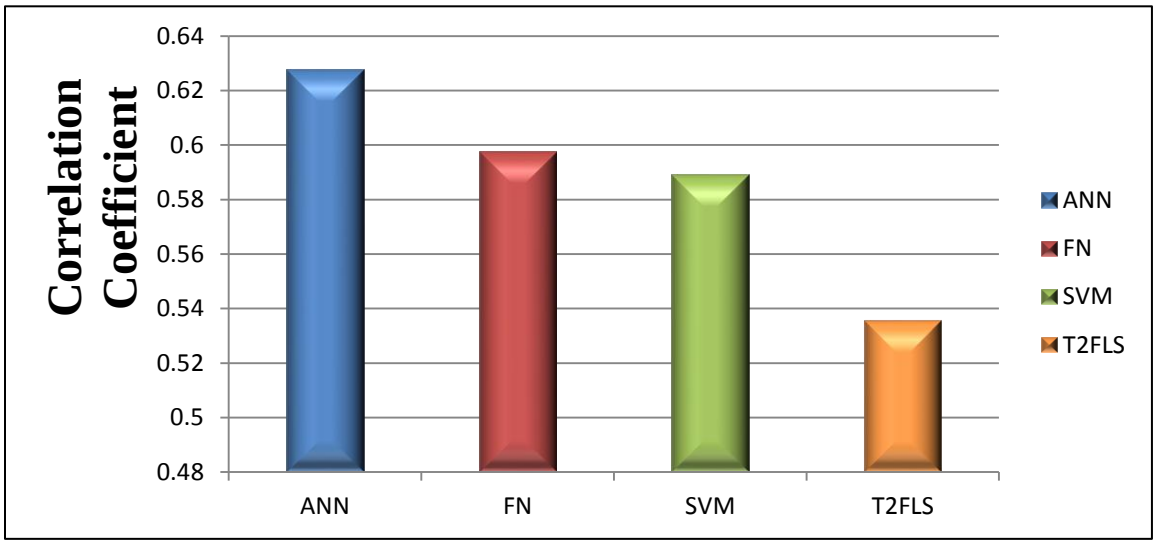


Figure 3-21 Correlation coefficient before Optimization

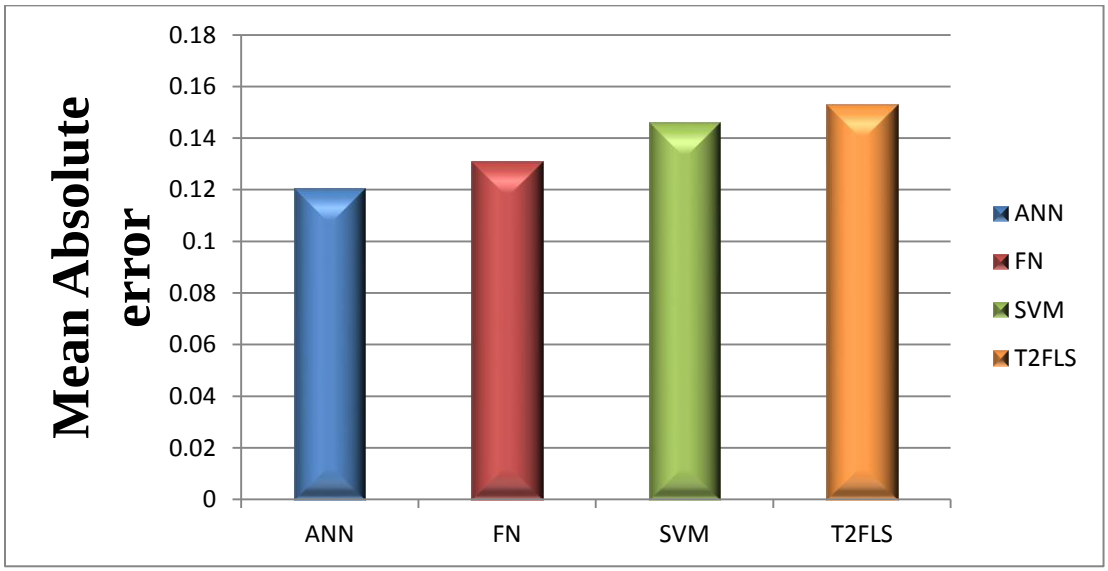


Figure 3-22 Mean absolute error comparison before Optimization

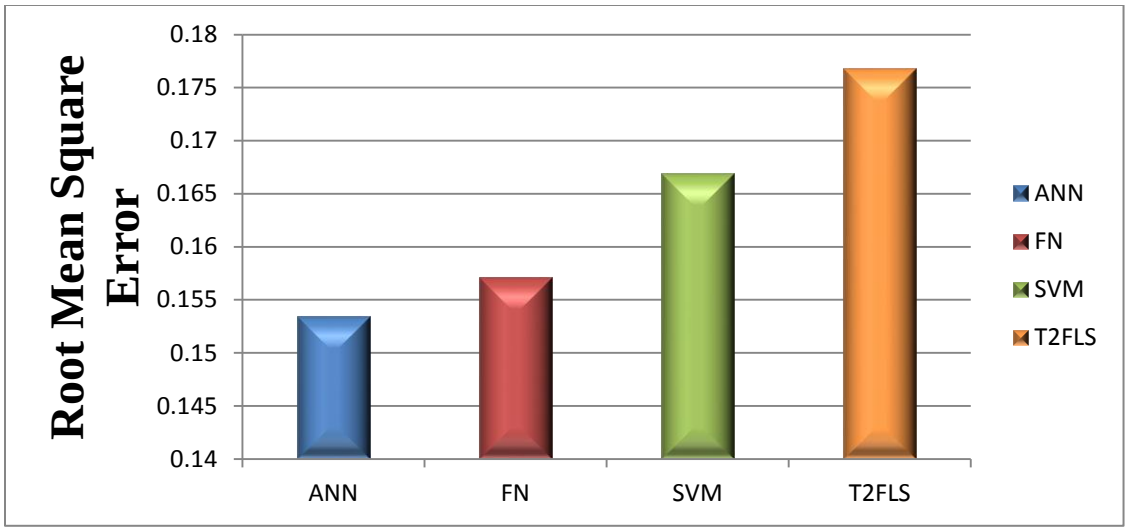


Figure 3-23 Root Mean Square Error before Optimization

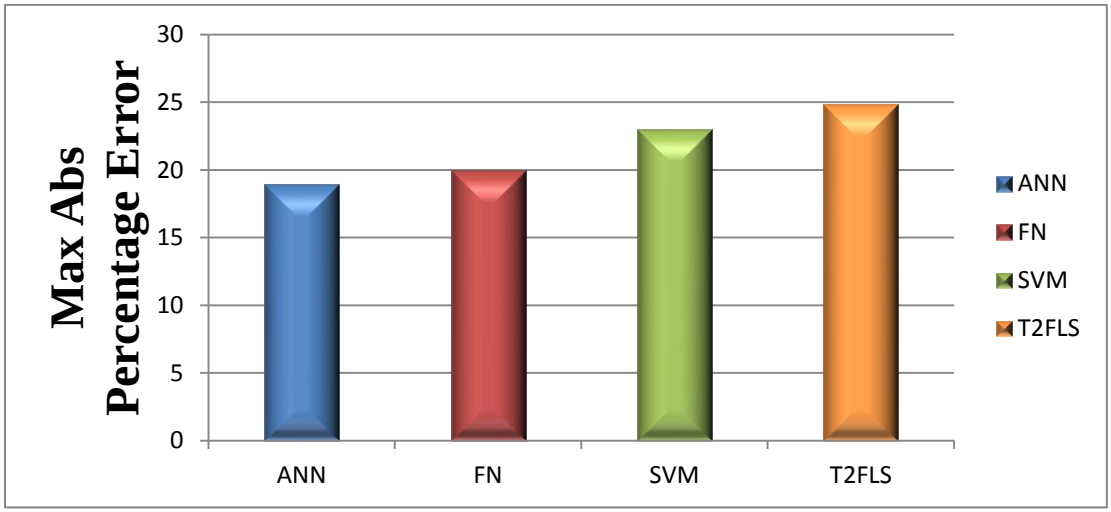


Figure 3-24 Maximum absolute percentage error before Optimization

3.5 Parametric Analysis

An individual parametric study was performed on each AI technique to enhance the model prediction capabilities. A small script inside matlab was developed to search for the combination of the optimum model parameters. Figures 3-25 through 3-28 show a range of possible minimum and maximum values for the model parameters during both training phase and testing phase. Then these model parameters were plotted against the correlation coefficient. The goal was to find where the model parameter overlap in both phases that correspond to the maximum correlation coefficient value. In the case of finding the optimum number of neuron in the ANN as in figure 3-25, it is clear that the optimum number of neuron is 6. There are other circumstances where the number of neuron is overlapping with the previous case yet such cases did not yield a better correlation coefficient. In Figure 3-26, the optimum value of the learning rate or alpha for the T2FSL is plotted. The value of alpha corresponding to that case is taken where the distance between the training and the testing phase is minimum. In the case of SVM there are two model parameters that were considered. They are the error allowance (lambda) and the regularization parameter, C. The optimum value of lambda is taken where its value in both the training phase and the testing phase is closest to each other (Figure 3-27). The value of C is chosen is a similar way, as shown in Figure 3-28.

Several cases were performed to arrive at the optimum values for these parameters which led to the optimum model for each of the AI techniques employed in this study. The prediction capability of these models is expected to improve and the results are presented in the following paragraphs.

Table 3-1 Summary of Optimized Parameters used in the models implementation

AI Model	Optimization Parameters
ANN	Number of hidden neurons =2 Number of neurons =6
SVM	Regularization Parameter, C=14000 Error allowance, Lambda= 0.0011 Penalty of Over fitting, epsilon=0.0007 Type of Kernel= polynomial Kernel Step Size =0.3 Verbose= 1
T2FLS	Learning rate, alpha=0.048

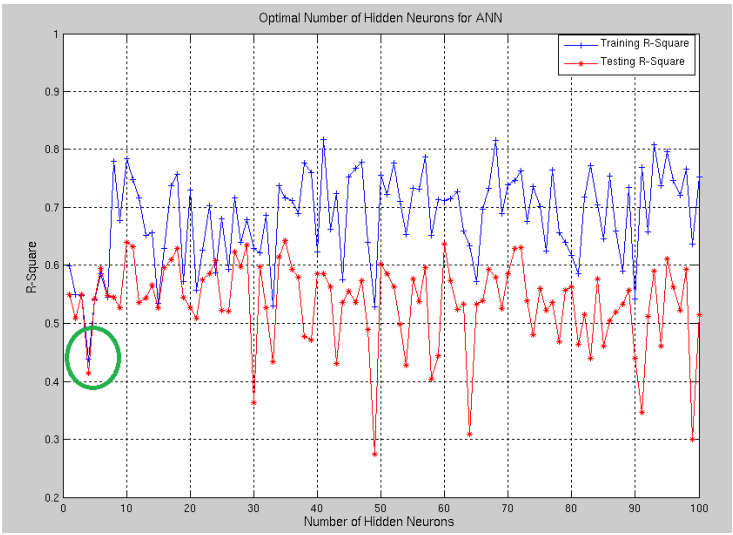


Figure 3-25 Optimal Numbers of Neurons

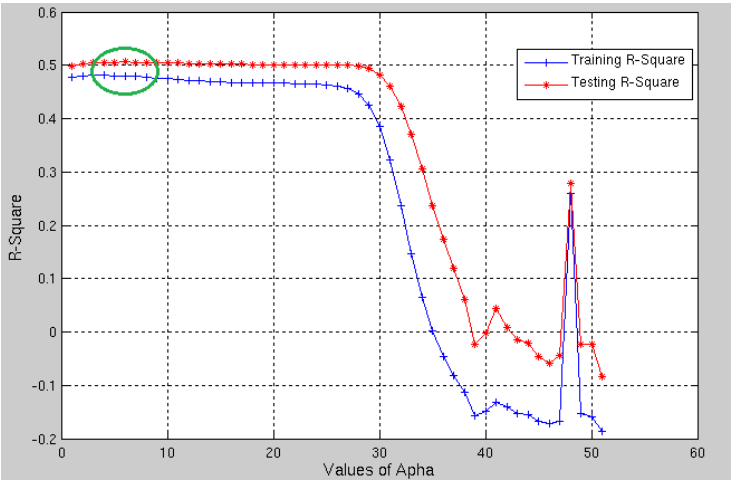


Figure 3-26 Optimal Value of Alpha for T2SLF

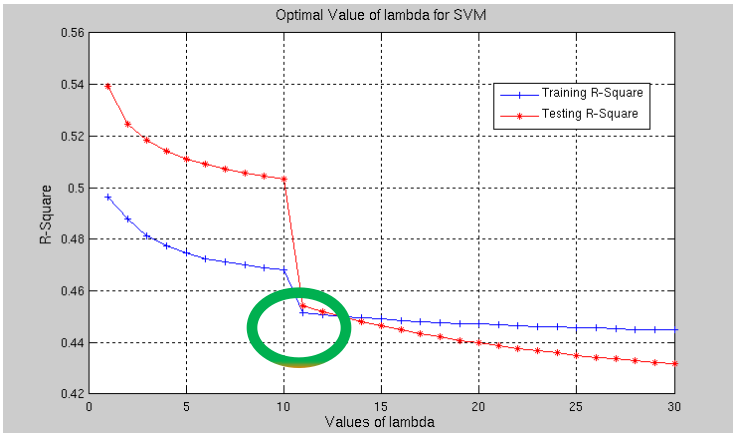


Figure 3-27 Optimal Value for Lambda for SVM

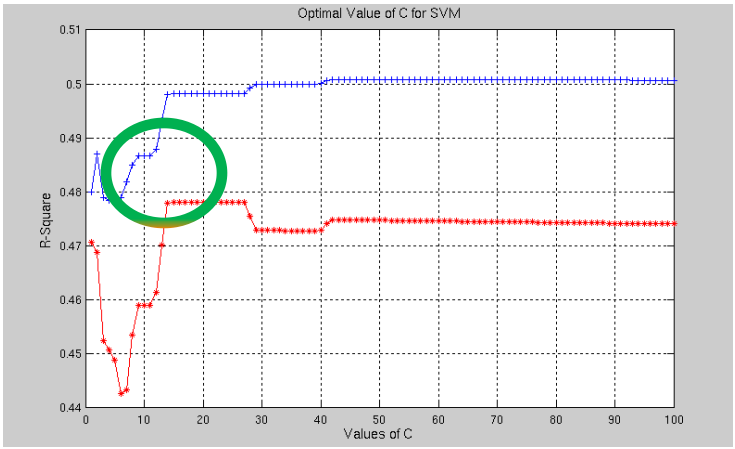


Figure 3-28 Optimal Value of C for SVM

Figures from 3-29 and 3-30 show results for the T2FLS after applying the optimized model parameters before running the prediction model. One can see from these figures that T2FLS was kept behaving the same way of poorly predicting water saturation and did not improve after applying the optimized model parameters. At the same time one can see that well#2 is in a better agreement with the model.

Figures 3-31 and 3-32 show the results for SVM. SVM too did not gain any substantial benefit of applying the optimized model parameters. Similarly as the T2FLSmodel, well#2 is in a better agreement with the model than well #1

Figures 3-33 and 3-34 show the results for the functional network. Compared to the results from previous models, FN showed some improvement as can be seen from the cross plot of the measure and predicted water saturation. The improvement was significant but not enough. However, one can see that both wells are relatively in agreement with the model.

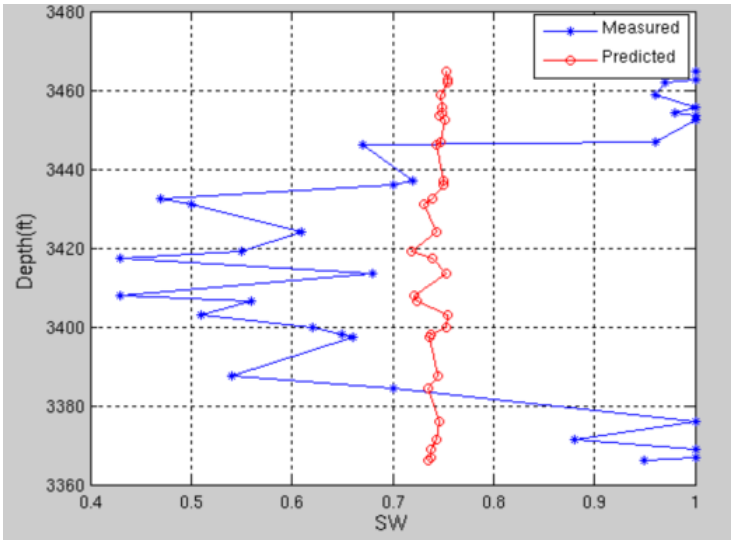


Figure 3-29a Depth vs Sw Testing using T2FLS Model for well#1

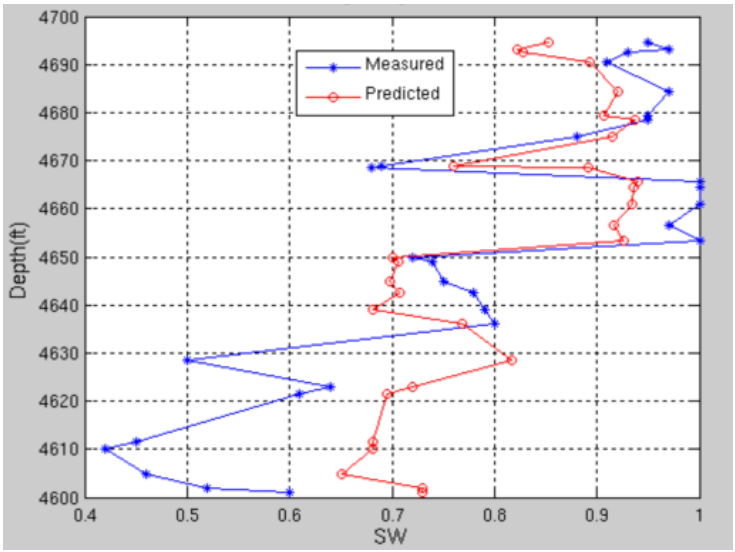


Figure 3-29b Depth vs Sw Testing using T2FLS Model for well#2

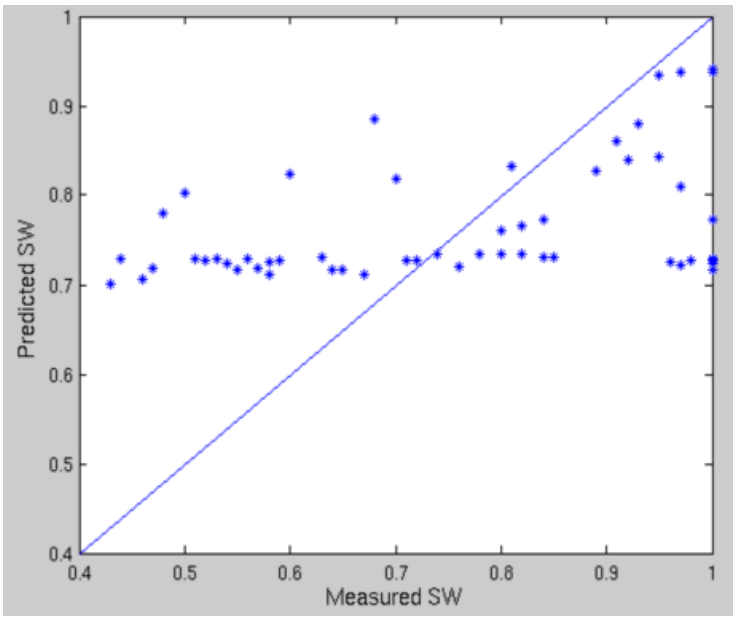


Figure 3-30 Crossplot of Measured and Predicted Sw (Testing) using T2FLS

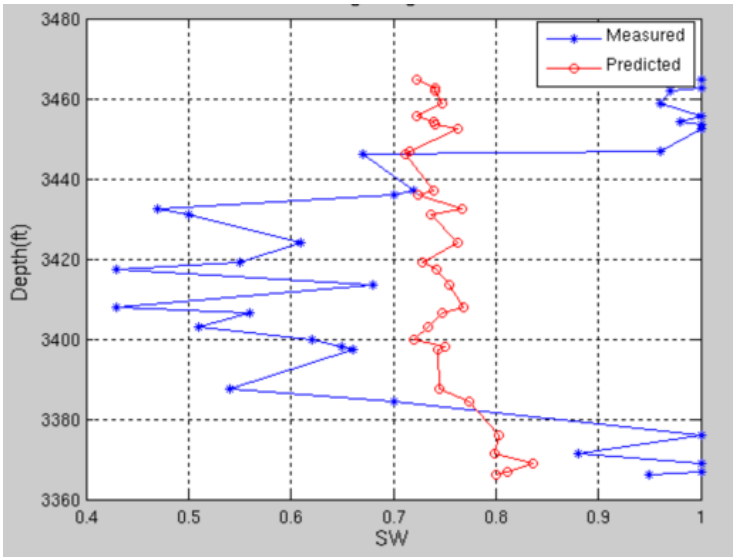


Figure 3-31a Depth vs Sw Testing using SVM Model for well#1

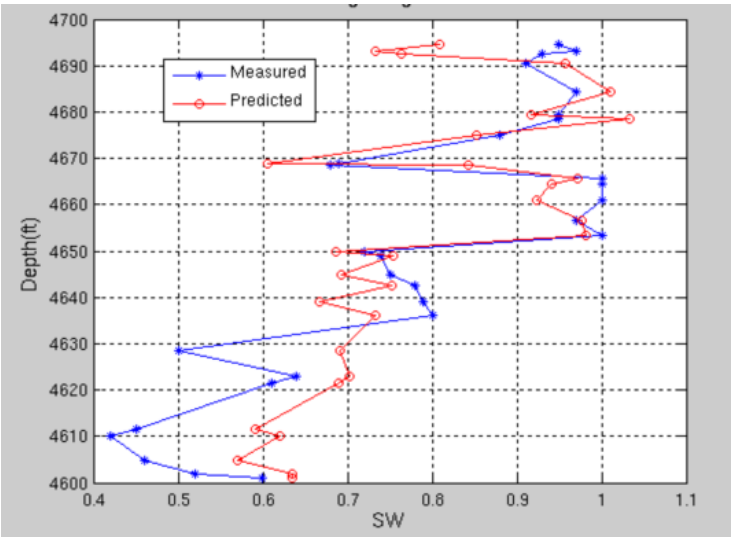


Figure 3-31b Depth vs Sw Testing using SVM Model for well#2

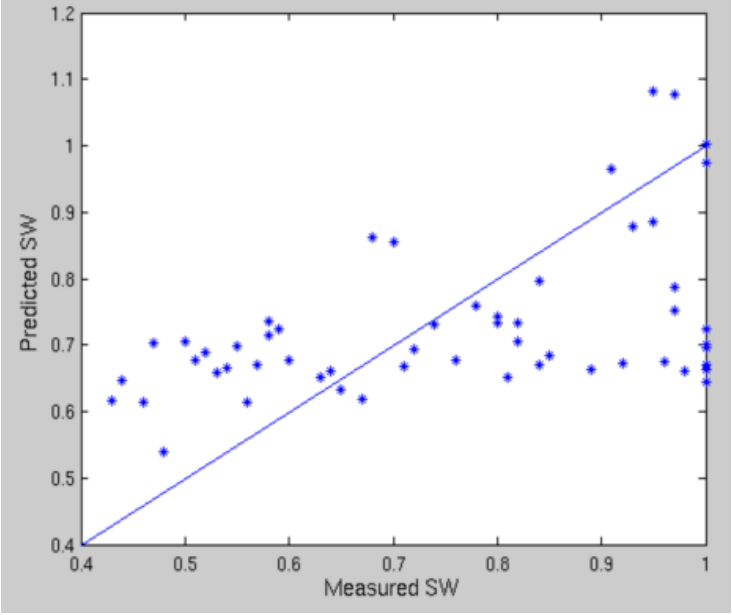


Figure 3-32 Crossplot of Measured and Predicted Sw (Testing) using SVM

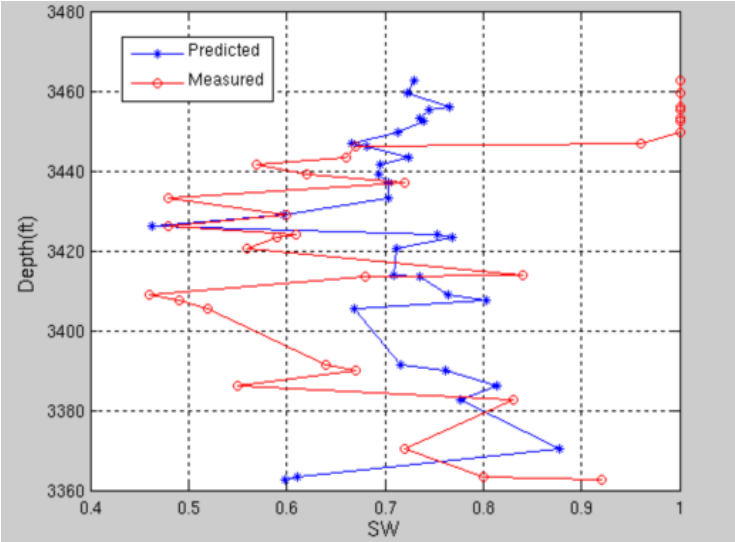


Figure 3-33a Depth vs Sw Testing using FN Model for well#1

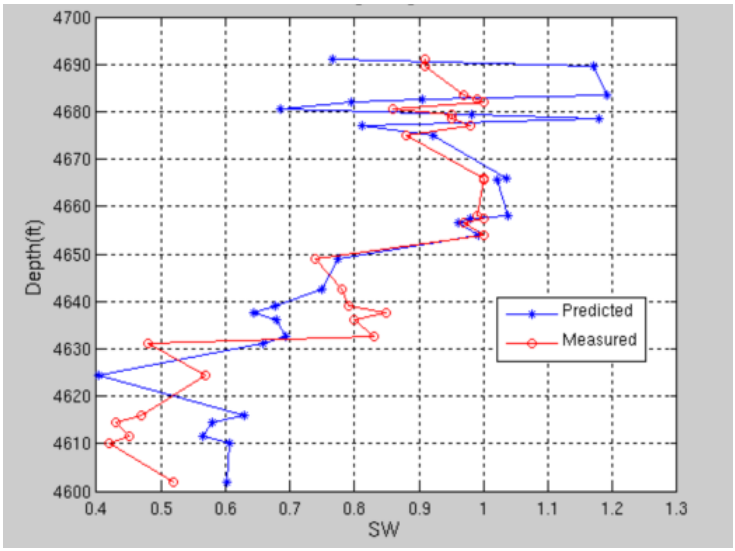


Figure 3-33b Depth vs Sw Testing using FN Model for well#2

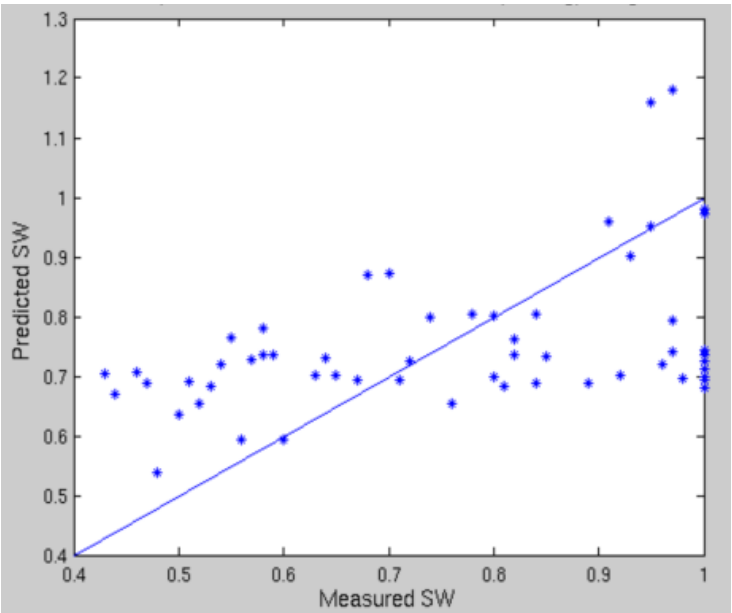


Figure 3-34 Crossplot of Measured and Predicted Sw (Testing) using FN

Finally, the Figures 3-35 and 3-36 show the results from the ANN model. It is clear by looking to the crossplots of the measured water saturation and the predicted values that the results lie within reasonable distance to the 45 degree line which indicate a good improvement of the prediction capabilities of ANN after applying the optimized model parameters. Similarly, as the functional network, both wells are relatively agreement with the model

Figures 3-37 through 3-40 are statistical representations of the results of all the AI models. It can be seen by looking at the statistical error charts that ANN has scored the highest correlation coefficient and the lowest score in the root mean square error, mean error and max percentage error. Also, one can notice that T2FLS has the lowest correlation coefficient but did poorly in the other statistical error measures.

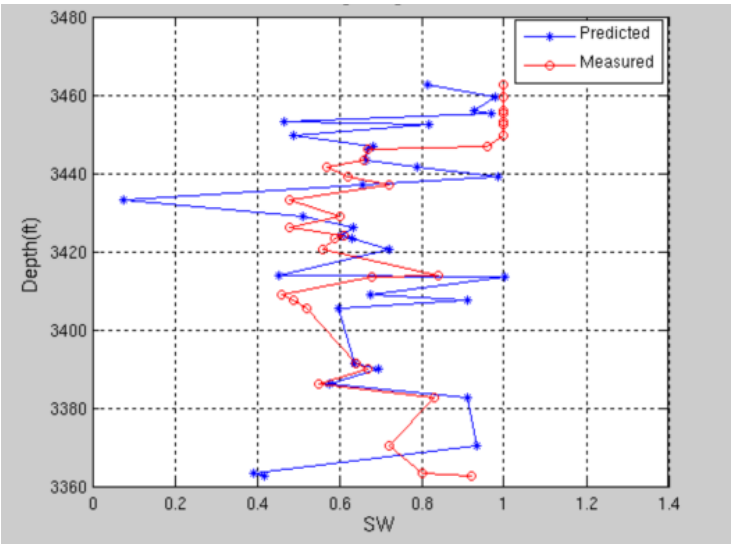


Figure 3-35a Depth vs Sw Testing using ANN Model for well#1

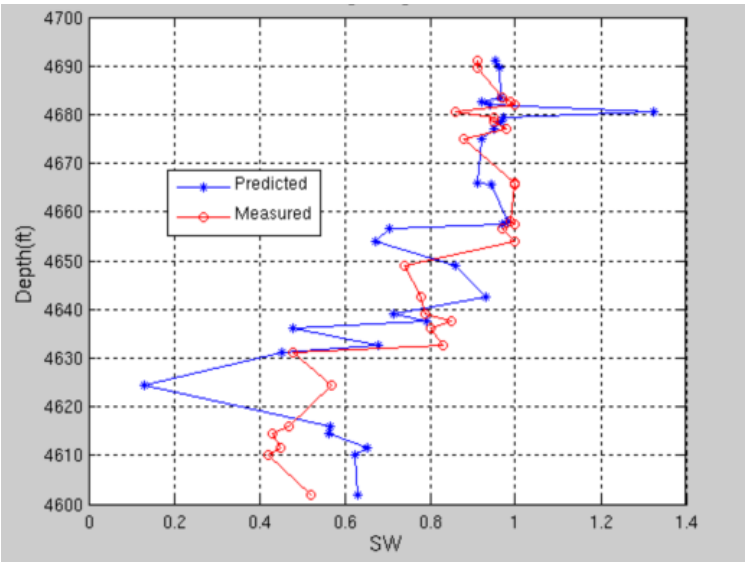


Figure 3-35b Depth vs Sw Testing using ANN Model for well#2

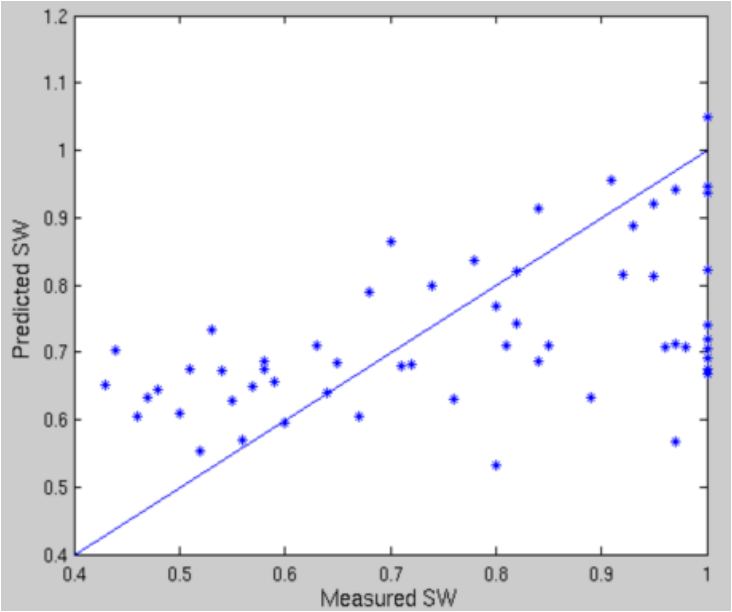


Figure 3-36 Crossplot of Measured and Predicted Sw (Testing) using ANN

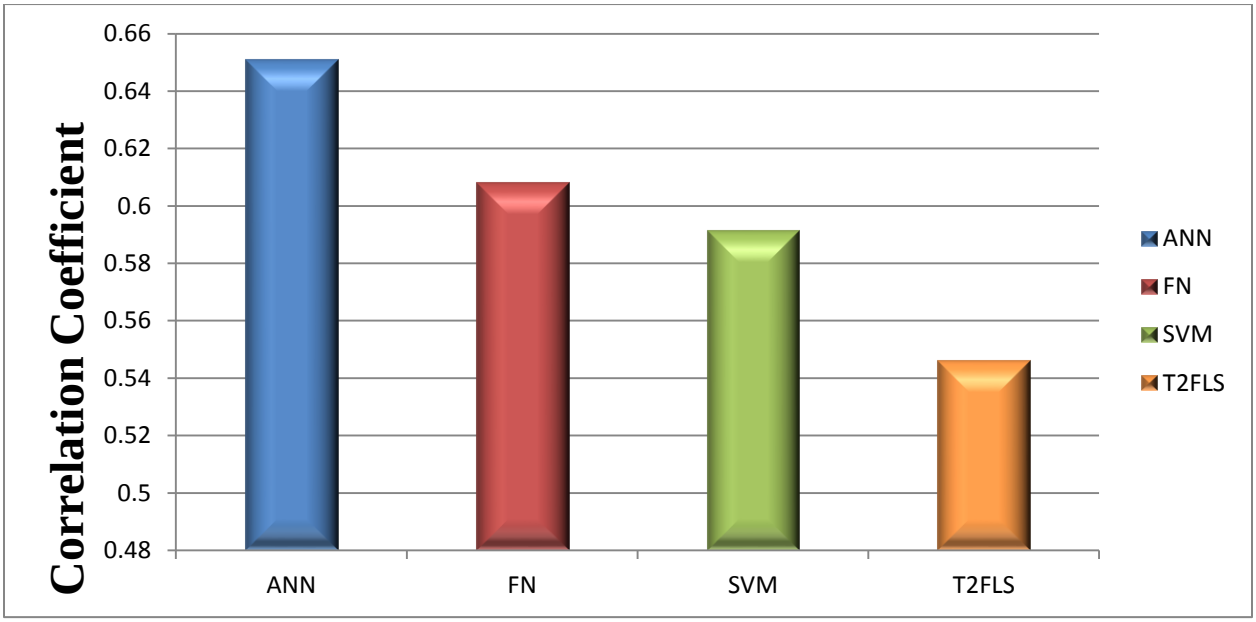


Figure 3-37 Correlation coefficient with optimized model parameters.

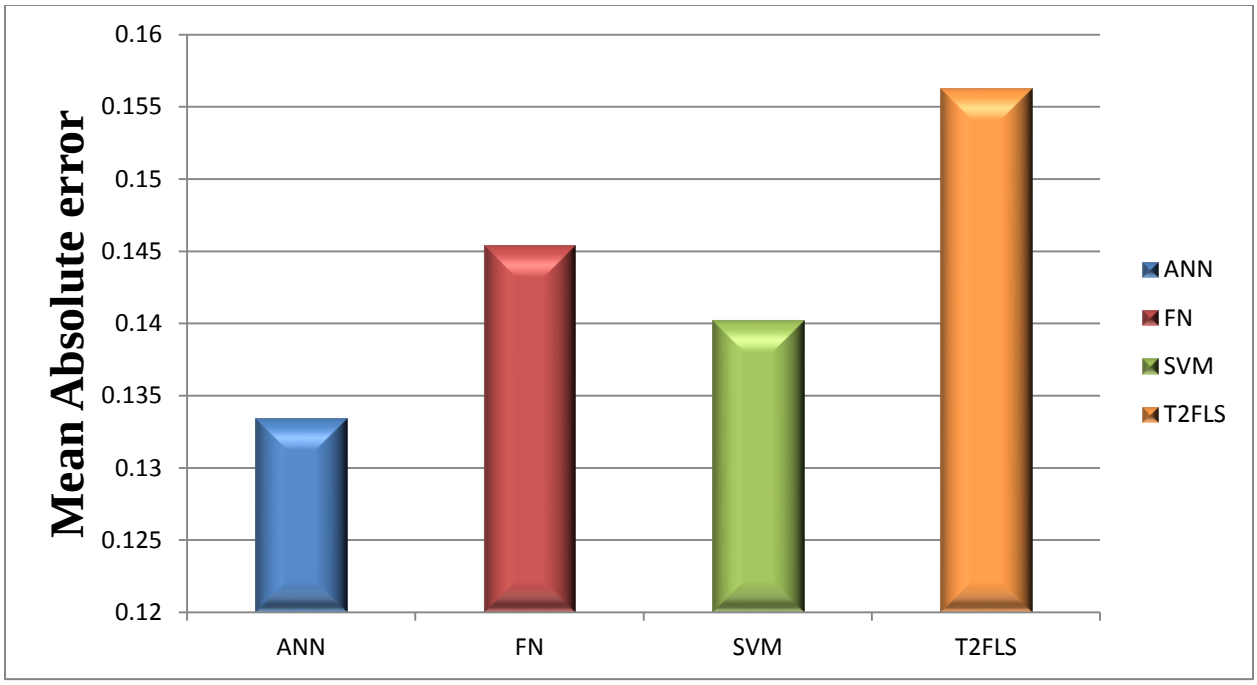


Figure 3-38 Mean absolute error with optimized model parameters

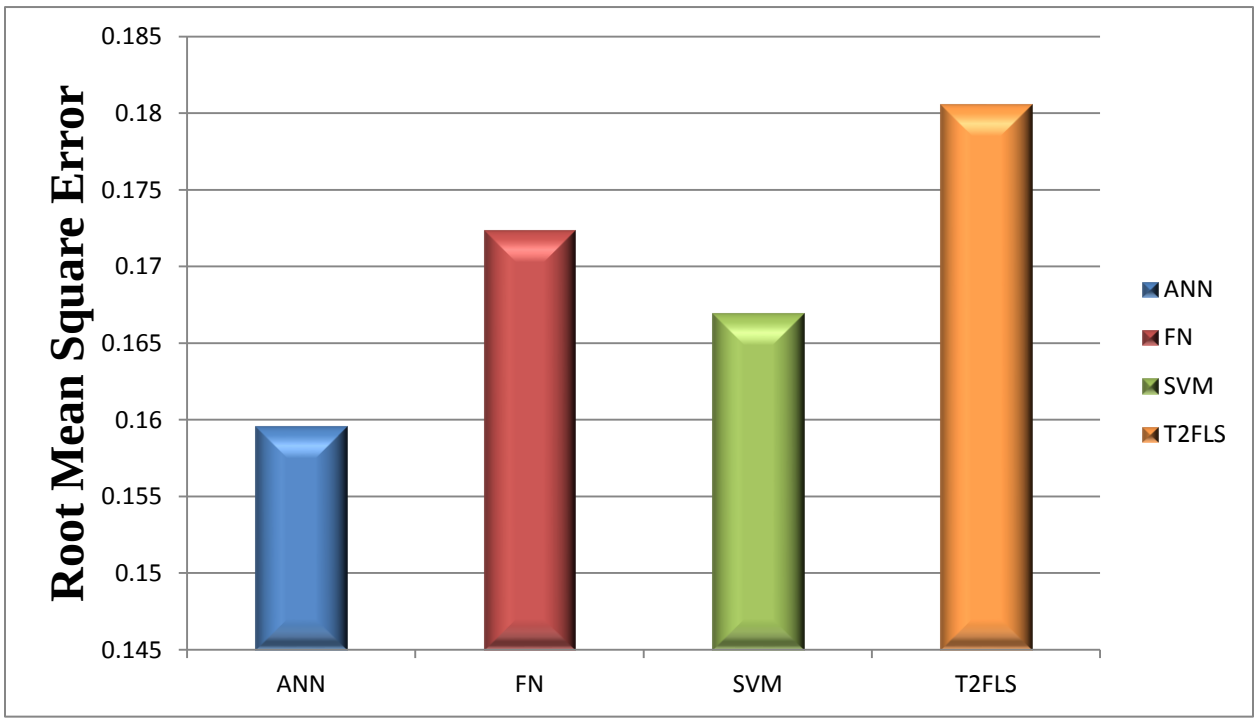


Figure 3-39 Root Mean Square Error With optimized model Parameters

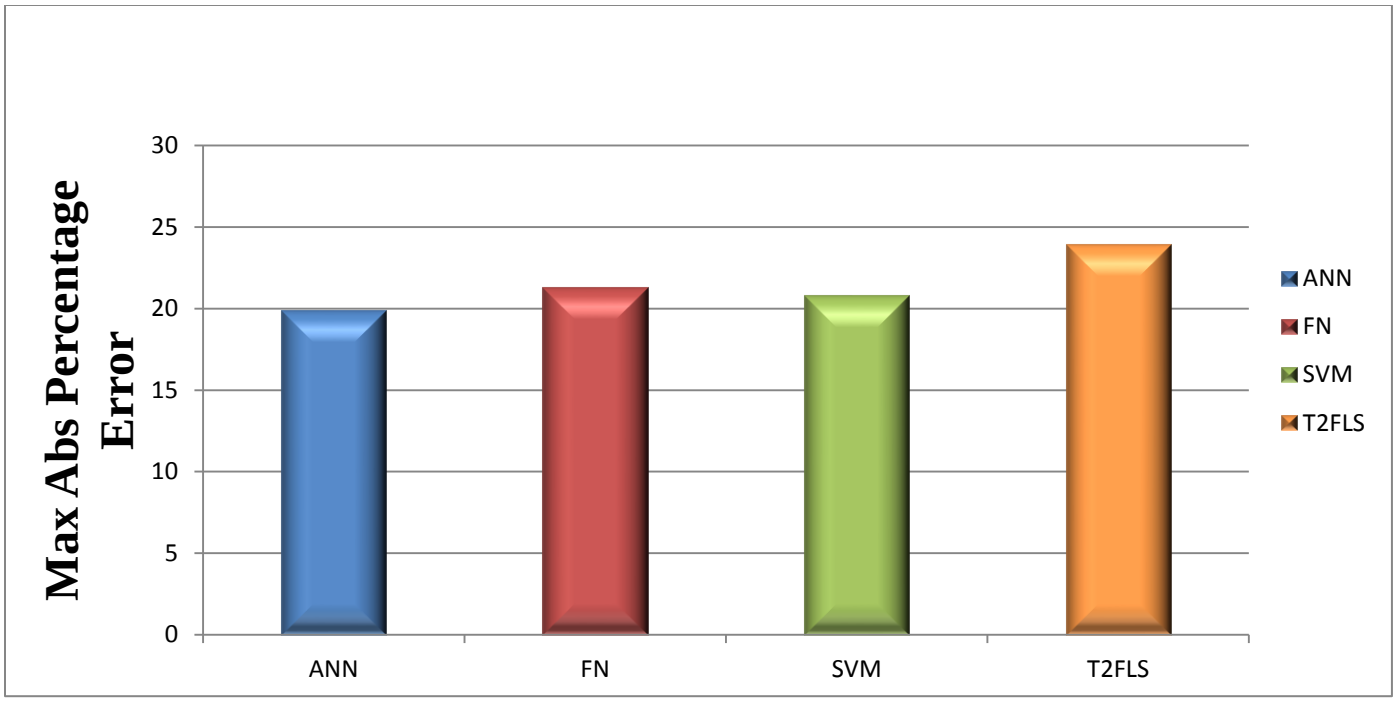


Figure 3-40 Maximum absolute percentage error with optimized model parameters

CHAPTER 4

Hybridization of AI Techniques

Hybridization of AI techniques is the procedure of integrating the outcome of several computational intelligence models to output only one technique. It has been reported in literature that AI modeling techniques has solved many different types of problems. These problems include missing data, planning, prediction and reasoning. Now as many other modeling techniques, different AI models have different limitation as well as different advantages. The role of AI hybridization comes here clear by leveraging on the individual AI advantages and to minimize individual AI weakness to reach to the optimum model.

In this chapter we will use two hybrid systems to find out the sensitivity of the input features in the data set and another hybrid system will be used to develop features selection based hybrid model.

4.1 Input Sensitivity Analysis

It is a well-known fact that AI models are data driven. However, putting all the available data as an input data does not guarantee the best AI model. We need to find out which one of these input data is contributing positively to the overall modeling and which ones are negatively contributing to the AI modelling. To address this issue the Multivariate Linear Regression Feature Selection system was used. This system gives the individual relationship between the input data and the output data by providing a correlation coefficient matrix for all the input data with the target data.

Table 4-1 shows the individual relationship between each feature used as an input with the target data. With a cutoff point of correlation value equal to 0.20 only three input data were chosen for all AI techniques. These input data are GR, PEF and RT.

Table 4-1 Multivariate Linear Regression Feature Selection

	A	B	C	D	E	F	G	H
1		*DT*	*GR*	*PEF*	*PHIE*	*RHOB*	*RT*	*SW_COR*
2	DT	1						
3	GR	0.372307	1					
4	PEF	0.081372	0.003614	1				
5	PHIE	0.538754	-0.28265	-0.35738	1			
6	RHOB	-0.68145	-0.14758	0.383644	-0.88999	1		
7	RT	-0.81413	-0.20472	-0.16196	-0.41569	0.485013	1	
8	SW_COR	-0.17583	0.26739	-0.22793	-0.08056	-0.0432	0.306412	1

Figures 4-1 and 4-2 show results for T2FLS. With the third experiment T2FLS behaved the same way as in the previous two experiments. The predicted water saturation was always above 0.7. This behavior can be seen clearly in the crossplot of the measured water saturation versus predicted water saturation. Furthermore, in the same plot it seems that there is a gap at higher water saturation where the model was not able to predict. Also, the measured water saturation and predicted water saturation plots versus depth do not show any satisfactory results particularly well#1. Figures 4-3 and 4-4 show the results for SVM. SVM too is giving poor prediction performance. All predictions are above the saturation value of 0.7. The same gap as was seen for T2FLS in the crossplot is observed for SVM. However, the measured water saturation and predicted water saturation plots versus depth was performing relatively better for well#2 with minimal improvement for well#1. Figures 4-5 and 4-6 show the results for the model using functional networks. At this experiment functional network was not doing as good as the

previous experiments where it was relatively better in prediction. This can be seen by the obvious dispersion of the points in the plot of the measured water saturation versus predicted water saturation. Again, the same gap as was seen for T2FLS and SVM in the crossplot is observed for FN. The measured and predicted water saturation plots versus depth are not satisfactory in this case.

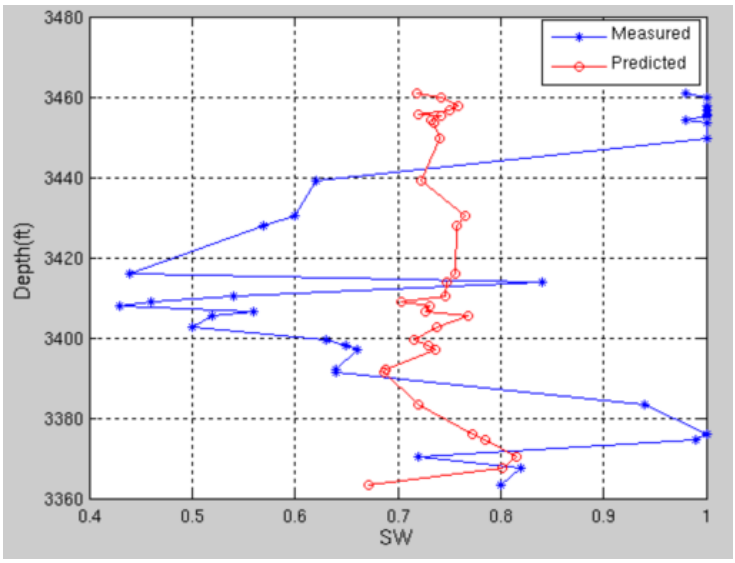


Figure 4-1a Depth vs Sw Testing using T2FLS Model for well#1

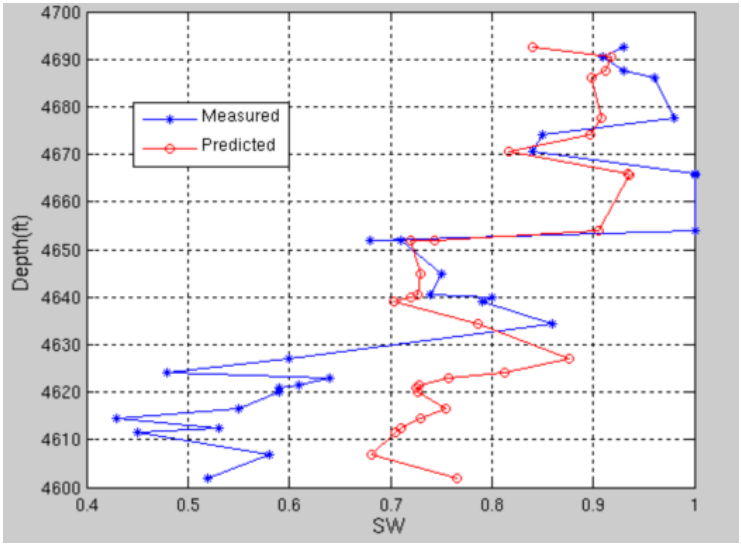


Figure 4-1b Depth vs Sw Testing using T2FLS Model for well#2

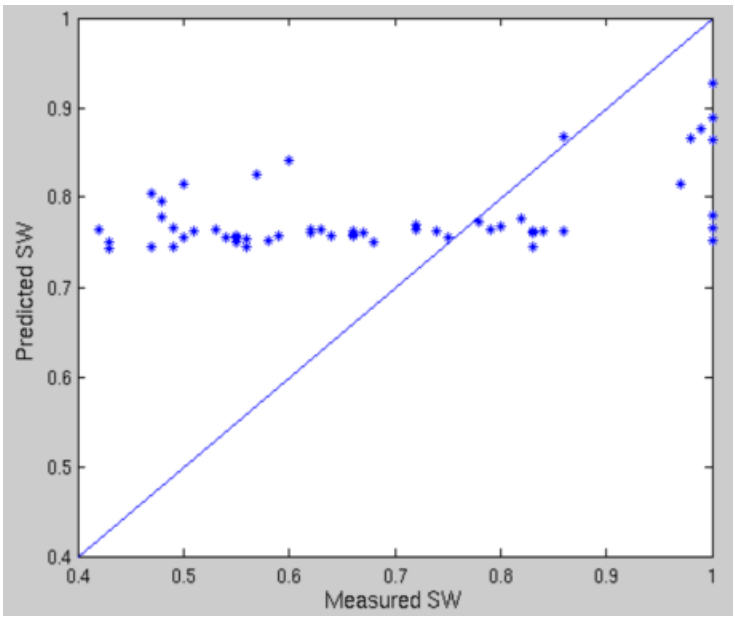


Figure 4-2 Crossplot of Measured and Predicted Sw (Testing) using T2FLS

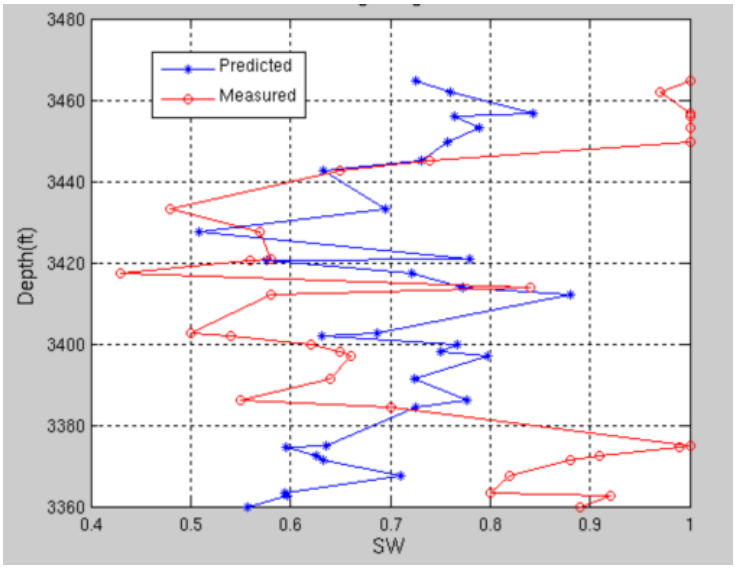


Figure 4-3a Depth vs Sw Testing using SVM Model for well#1

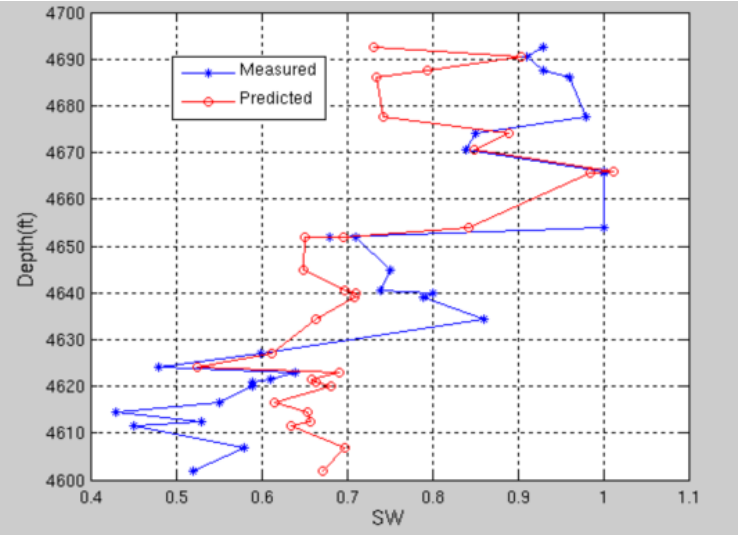


Figure 4-3b Depth vs Sw Testing using SVM Model for well#2

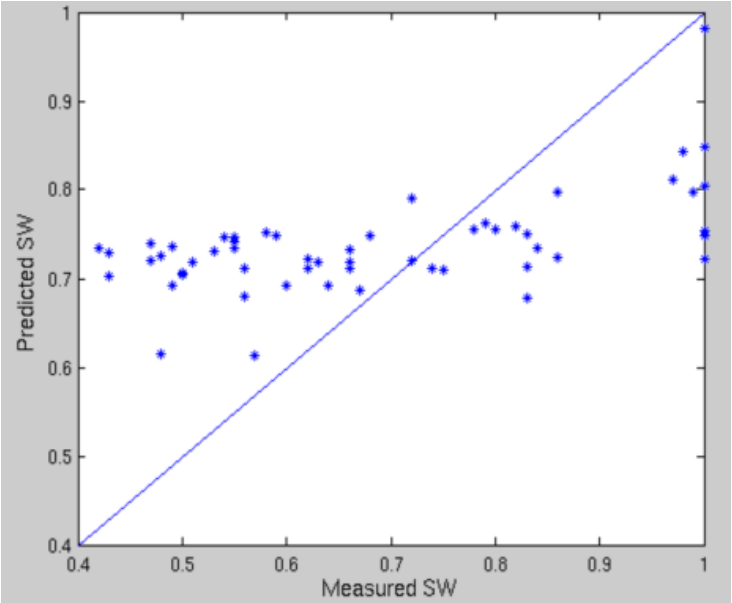


Figure 4-4 Crossplot of Measured and Predicted Sw (Testing) using SVM

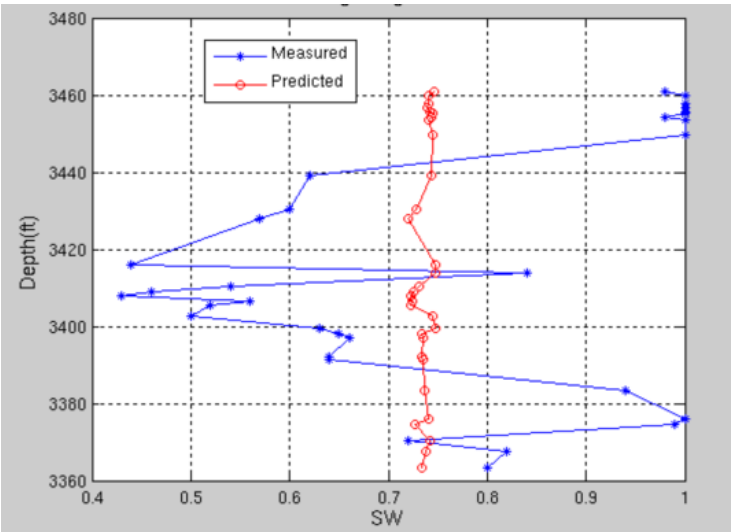


Figure 4-5a Depth vs Sw Testing using FN Model for well#1

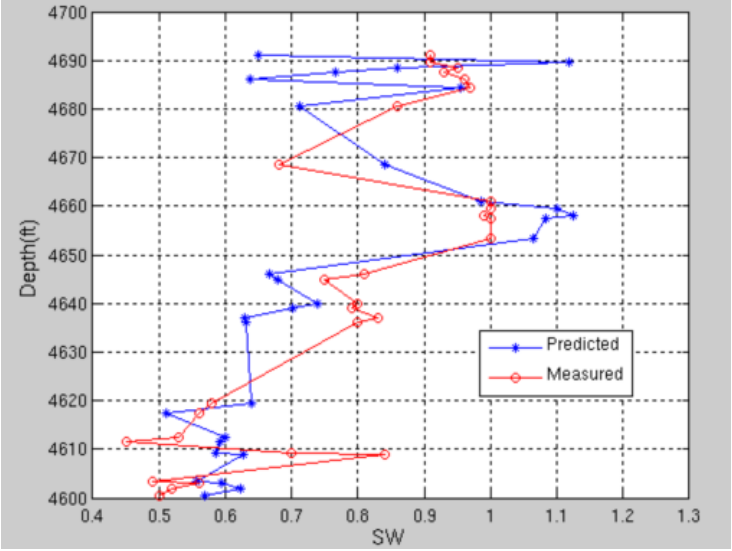


Figure 4-5b Depth vs Sw Testing using FN Model for well#2

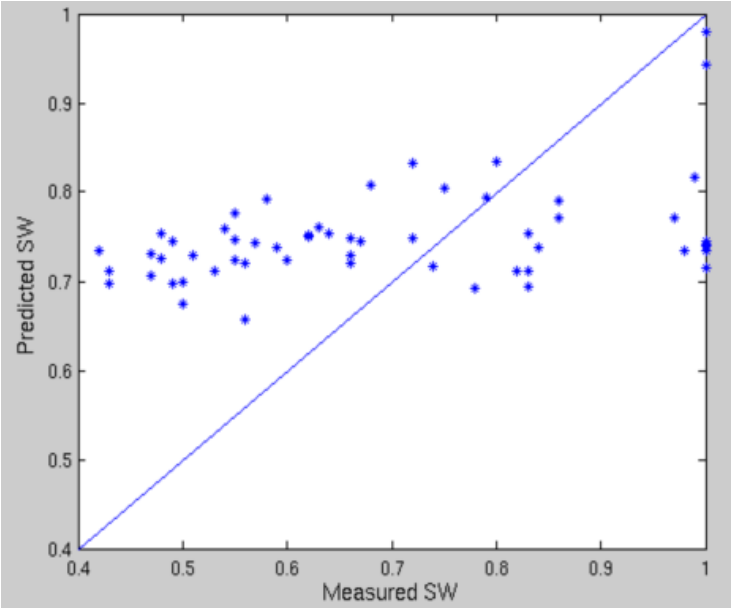


Figure 4-6 Crossplot of Measured and Predicted Sw (Testing) using FN

Figures 4-7 and 4-8 show the results for the ANN. At this experiment ANN performance was degraded comparing to its performance at the previous two experiments. It started to behave as T2FLS with respect to its predictions which were always above 0.7. Also, the measured water saturation and predicted water saturation plots versus depth for both wells were relatively better than the previous models.

It seems that all the AI models did not gain any value by applying the multivariate linear regression on the input dataset. Figures 4-9 through 4-12 show the results in terms of statistical performance of the AI techniques. T2FLS took the lead by scoring the highest correlation coefficient. This is due to the good agreement of well#2 with the model. As for the remaining statistical error measurements, ANN performed well by scoring low values for the root mean square error, the mean absolute error and the maximum absolute percentage error despite the fact that ANN did not score the highest correlation coefficient compared to others. Functional network performed the worst with respect to correlation coefficient as well as other statistical error measurements.

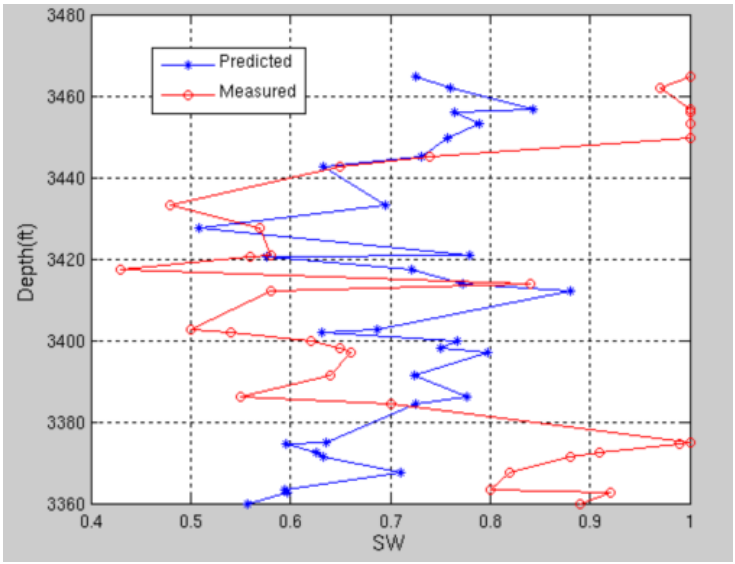


Figure 4-7a Depth vs Sw Testing using ANN Model for well#1

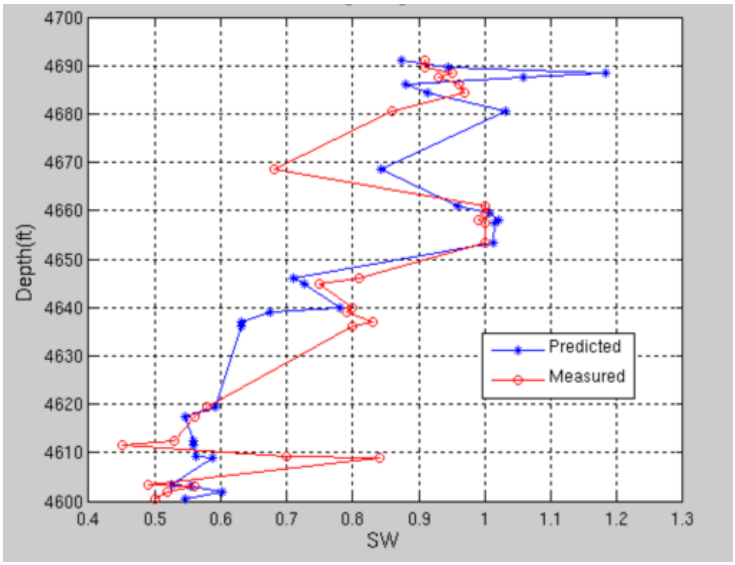


Figure 4-7a Depth vs Sw Testing using ANN Model for well#1

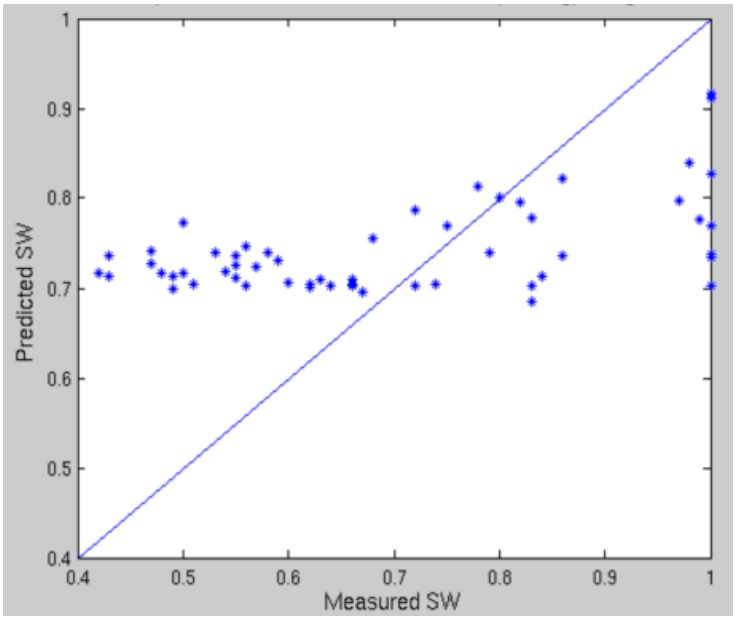


Figure 4-8 Crossplot of Measured and Predicted Sw (Testing) using ANN

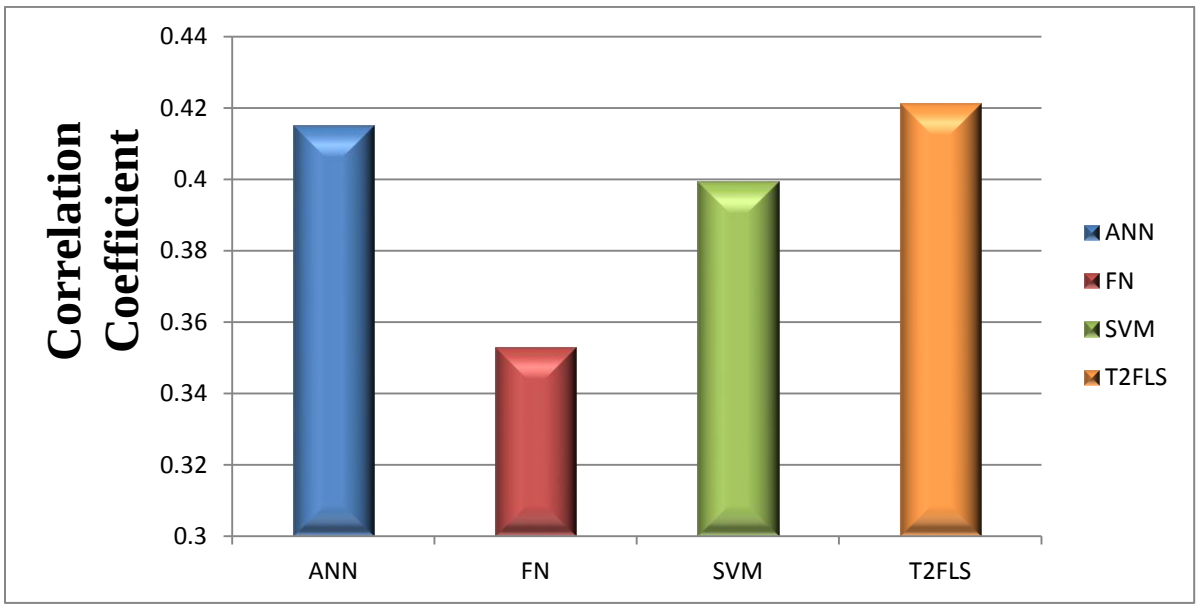


Figure 4-9 Correlation coefficient with Sensitivity Analysis Input

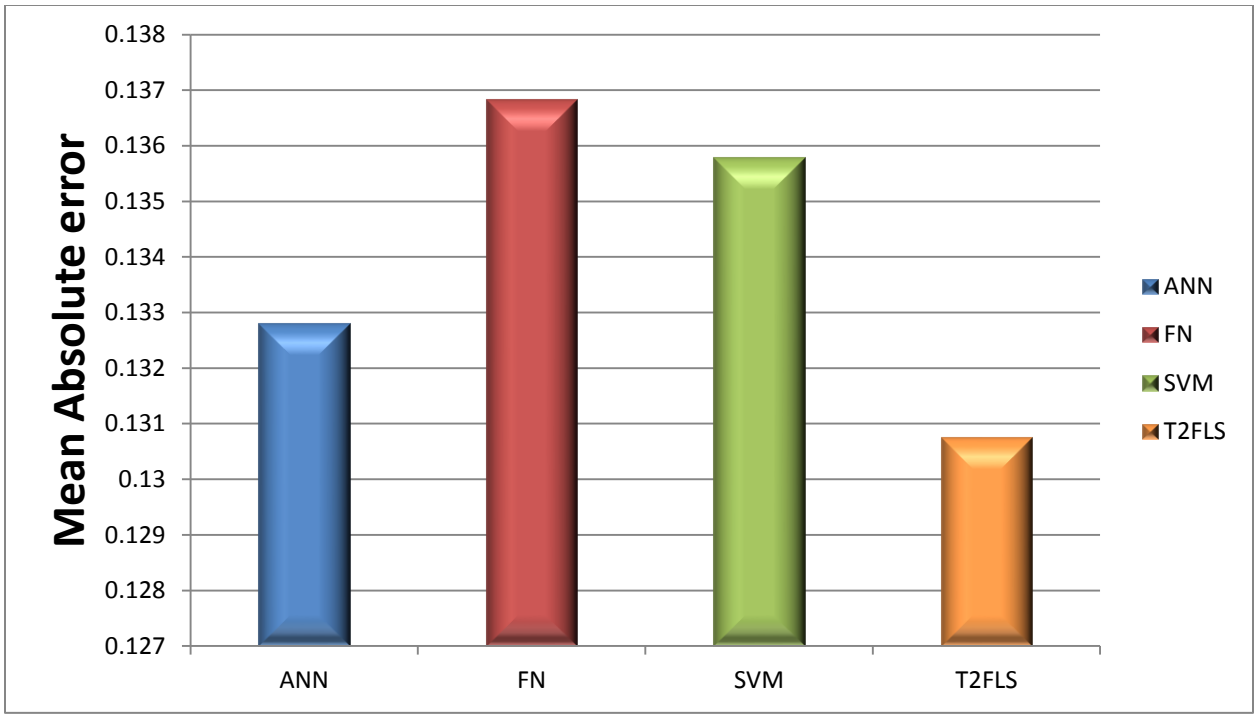


Figure 4-10 Mean absolute error with Sensitivity Analysis Input

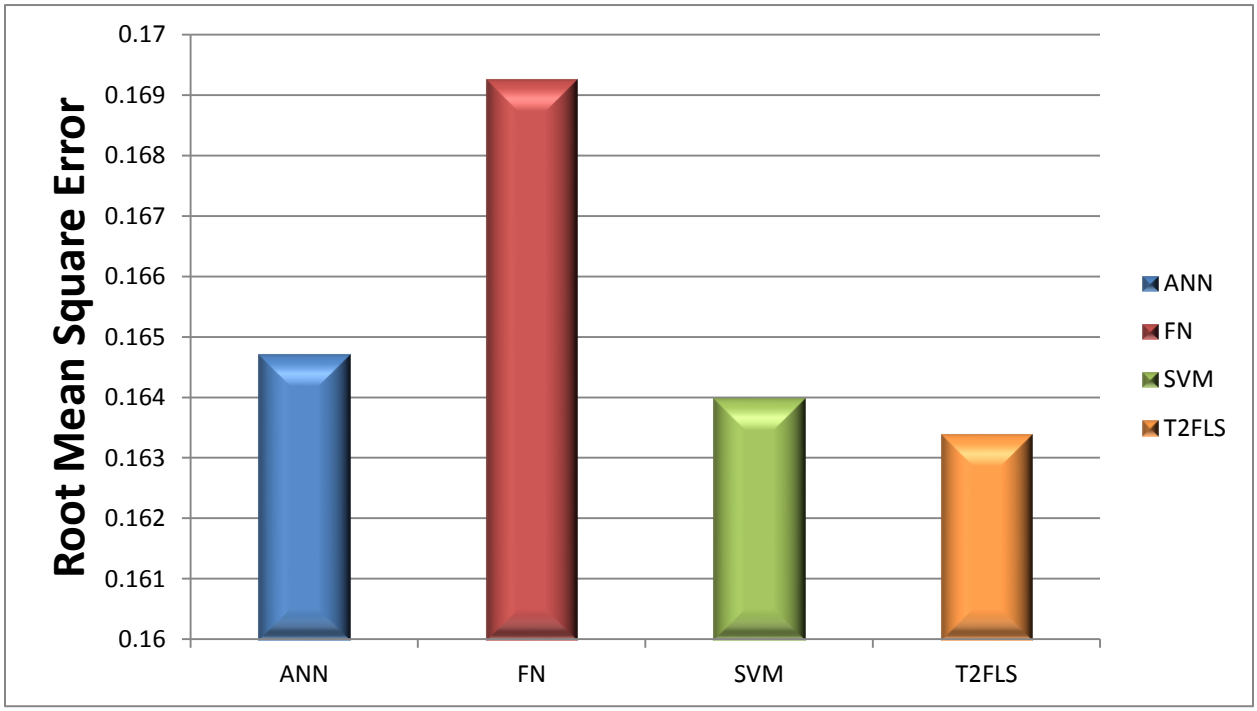


Figure 4-11 Root mean square with Sensitivity Analysis Input

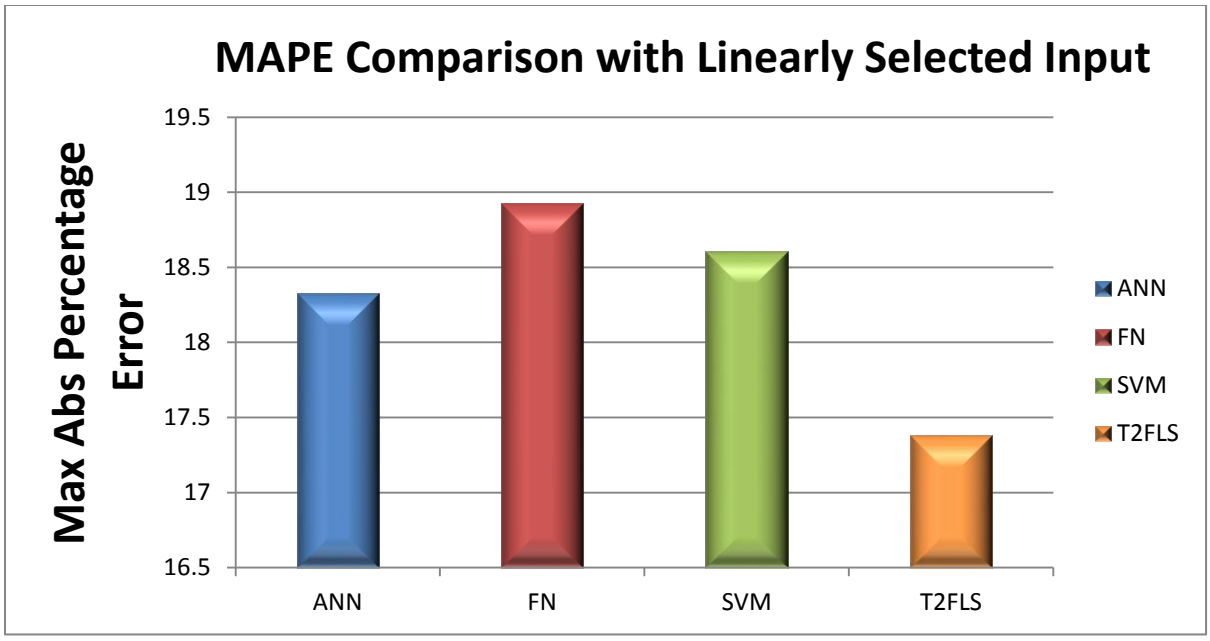


Figure 4-12 Maximum absolute percentage error with Sensitivity Analysis input

4.2 Feature Selection Based Hybrid Model

The increased popularity of the hybrid schemes is due to the extensive success in many real-world complex problems. Our motivation for this implementation includes the search for the higher performance accuracy in the prediction of water saturation property in carbonate reservoirs. Different AI models have their own shortcoming. One approach to over these shortcomings is by implementing hybridization of these AI models. The value added of the hybridization of different AI models together is to take advantage of the strength of both AI models. When literature review was conducted, there were may papers and journals recommended the utilization of AI hybridization. Several of these publications have demonstrated the advantages of these AI hybridization over the conventional AI techniques in terms of robustness and increase of prediction capabilities. Such publication includes (Anifowose and Abdulraheem, 2010). In this section we are using the functional network and the artificial neuron network hybrid system. This hybrid system has output the best subset input data namely GR, PHIE and Rt. After that, AI techniques were run using this best subset data.

Figure 4-13 and 4-14 show the result of T2FLS. As in the previous experiments T2FLS model predicted values which were above the saturation value of 0.7 for all the cases. Nevertheless, it improved its prediction capabilities when compared to the previous experiments. The measured predicted water saturation versus depth has improved to some extent for both wells.

Figures 4-15 and 4-16 show the results of SVM. The crossplot of the measured water saturation and the predicted water saturation show good distribution of the predicted values and the measured values over the 45 degree line. The water saturation versus depth show relatively better agreement between the measured and predicted values for both wells. Figures 4-17 and 4-18 show the results for functional network. FN was performing quite well in this experiment.

As for the measured water saturation and the predicted water saturation plots versus depth, well#2 was in a better agreement with the model.

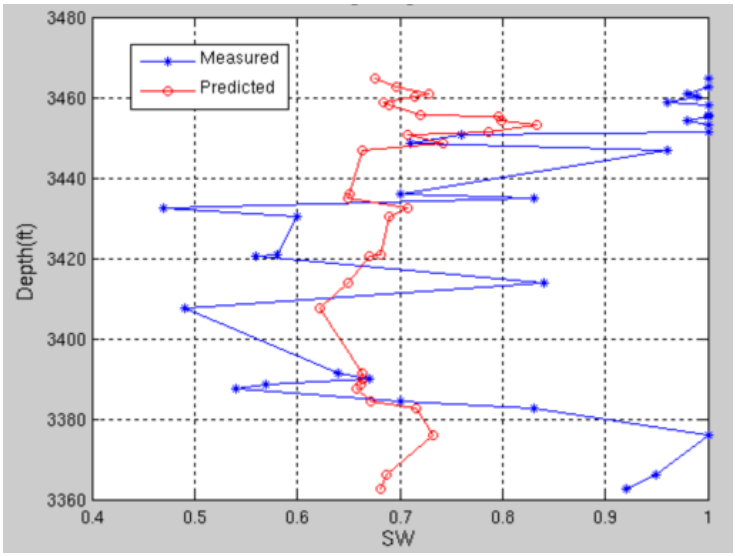


Figure 4-13a Depth vs Sw Testing using T2FLS Model for well#1

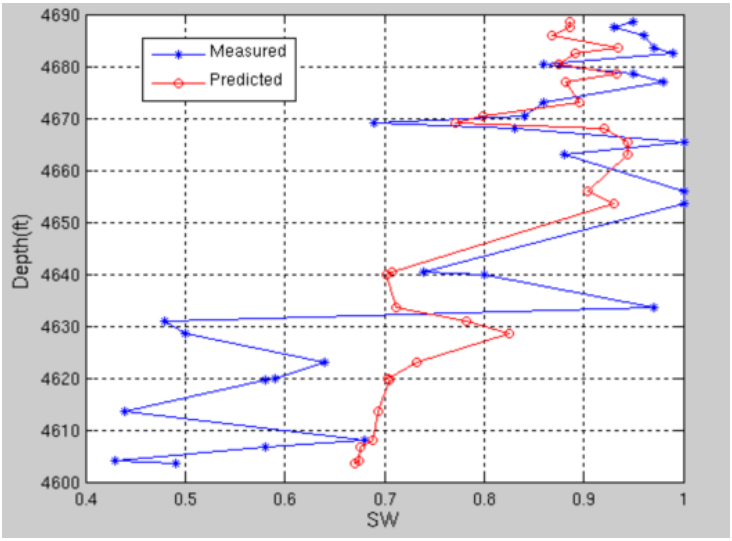


Figure 4-13b Depth vs Sw Testing using T2FLS Model for well#2

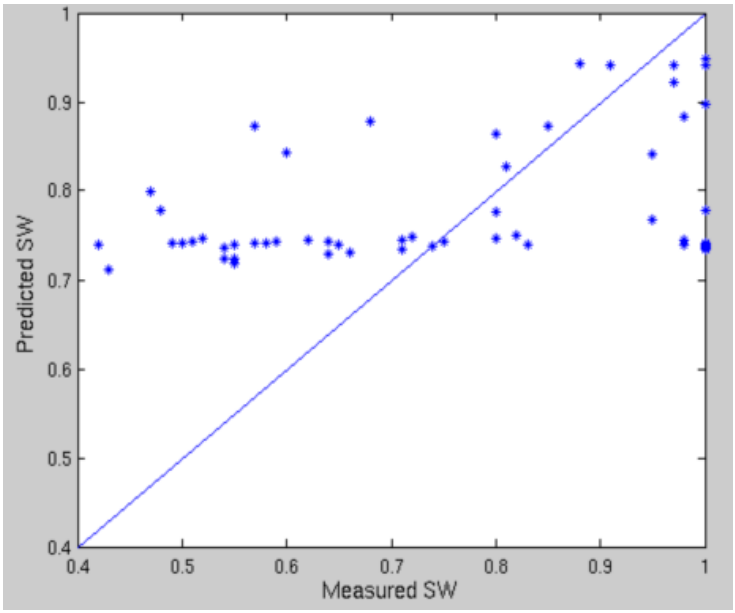


Figure 4-14 Crossplot of Measured and Predicted Sw (Testing) using T2FLS

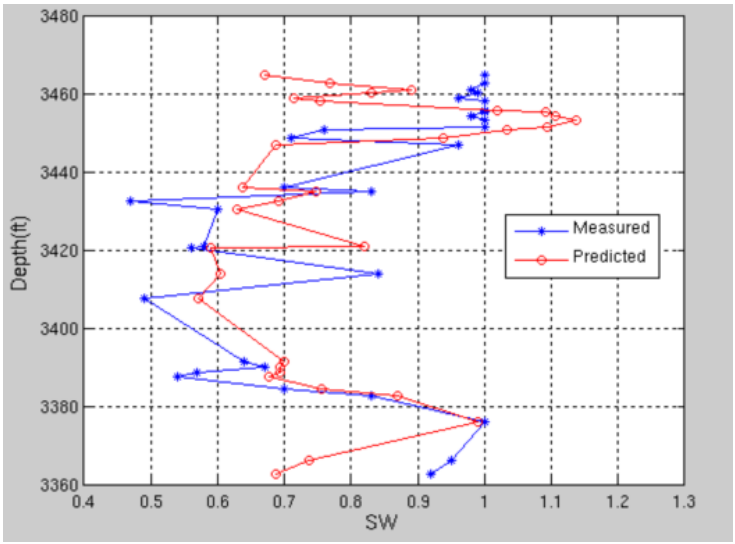


Figure 4-15a Depth vs Sw Testing using SVM Model for well#1

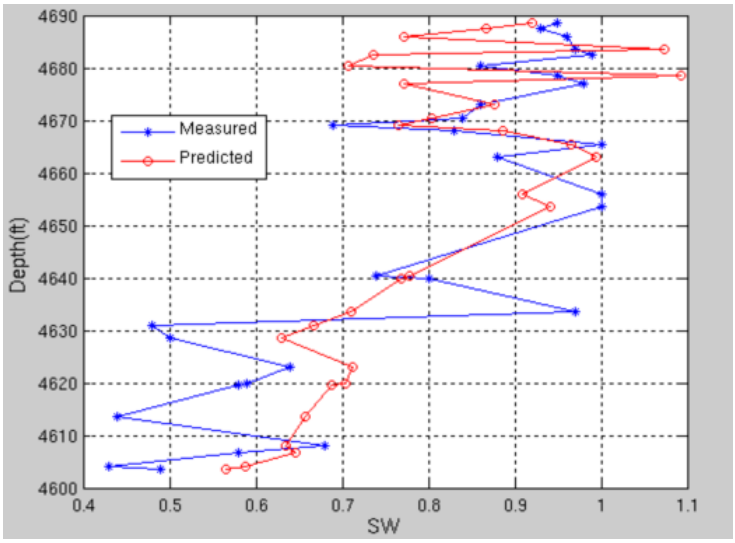


Figure 4-15b Depth vs Sw Testing using SVM Model for well#2

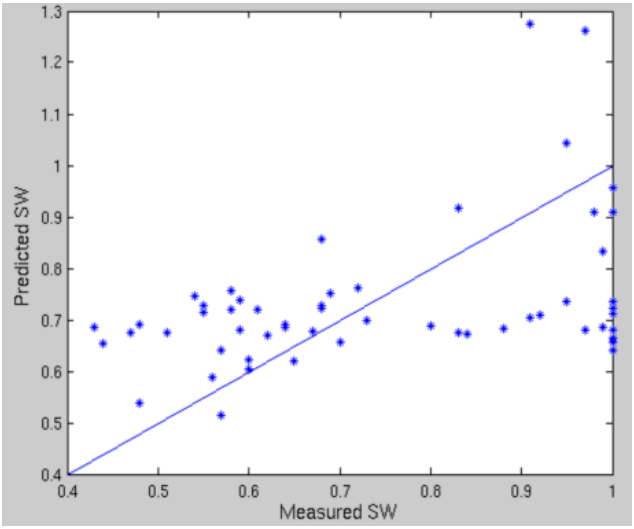


Figure 4-16 Crossplot of Measured and Predicted Sw (Testing) using SVM

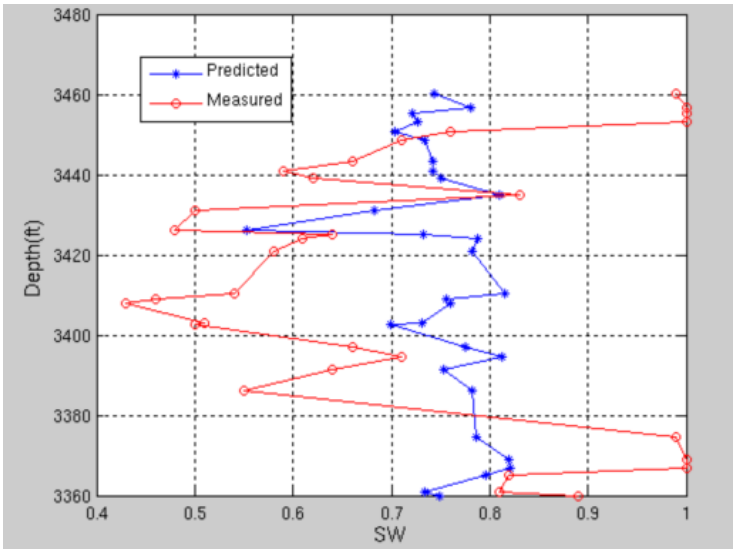


Figure 4-17a Depth vs Sw Testing using FN Model for well#1

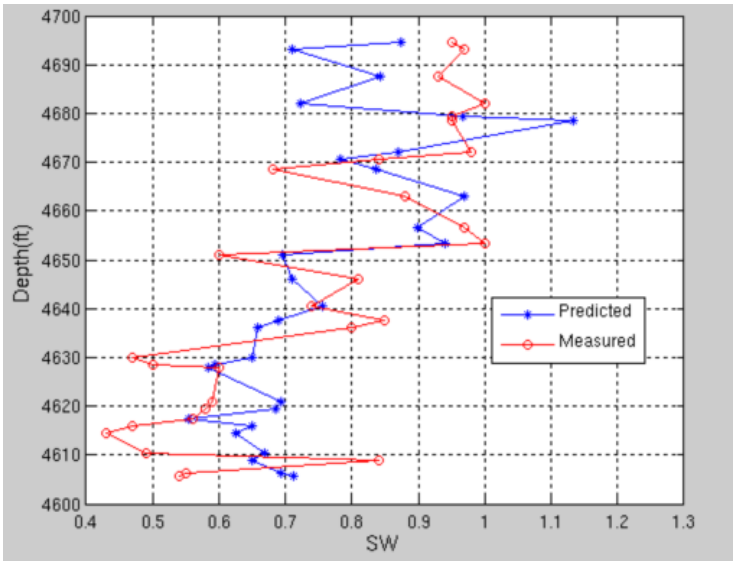


Figure 4-17b Depth vs Sw Testing using FN Model for well#2

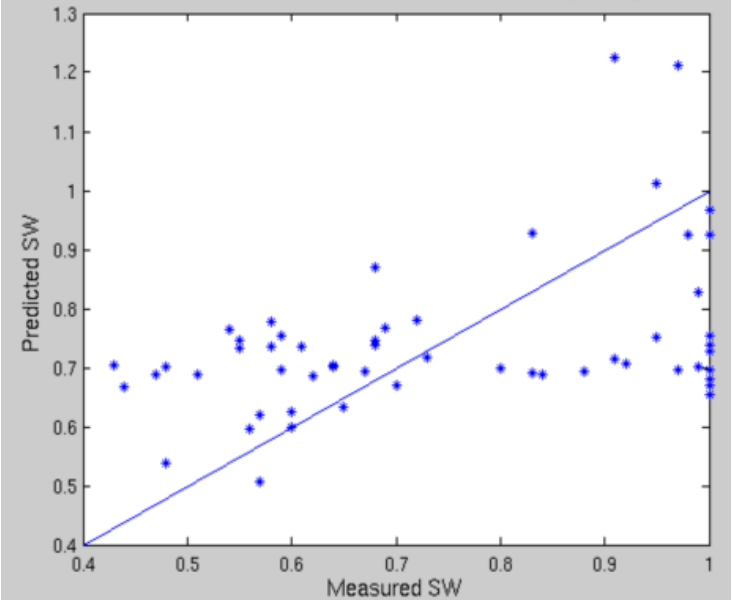


Figure 4-18 Crossplot of Measured and Predicted Sw (Testing) using FN

Figures 4-19 and 4-20 show the results of the ANN. Looking at the different crossplots one can see that ANN was performing better when compared to other AI technique. As for the measured water saturation and the prediction values versus depth, one can see that both wells are in a good agreement with the model

Figures 4-21 through 4-24 show the result of the statistical error measurements. ANN was leading by outperforming the other AI techniques in all of statistical error. Also T2FLS was the worst performing AI technique in this experiment with the lowest correlation coefficient and the highest of the RMSE, MAE and MAPE

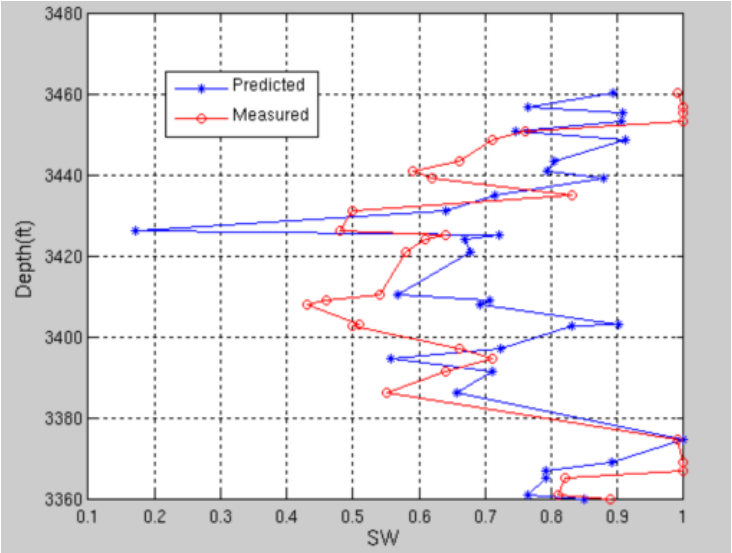


Figure 4-19a Depth vs Sw Testing using ANN Model for well#1

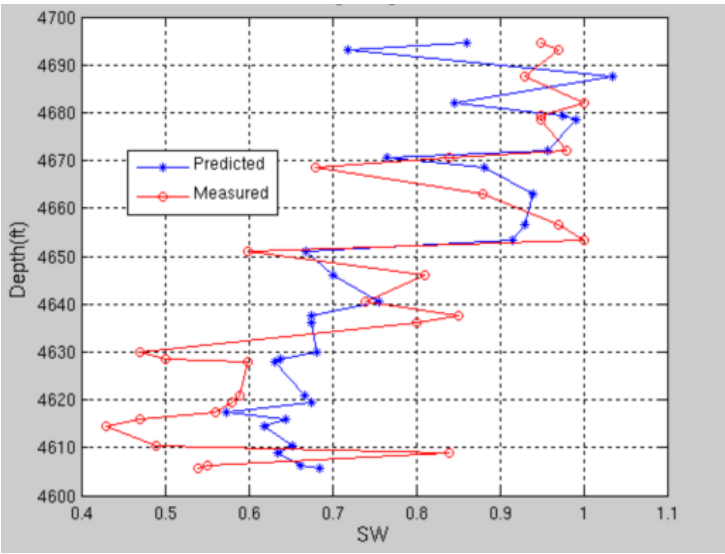


Figure 4-19b Depth vs Sw Testing using ANN Model for well#2

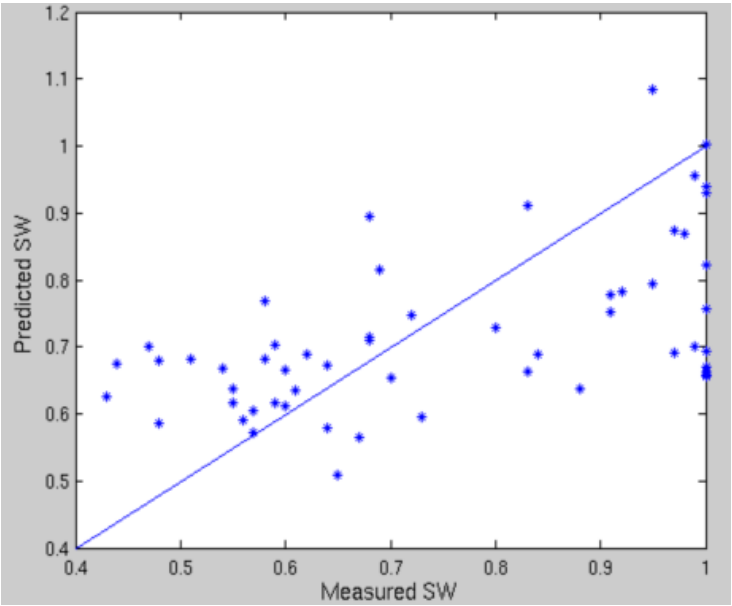


Figure 4-20 Crossplot of Measured and Predicted Sw (Testing) using ANN

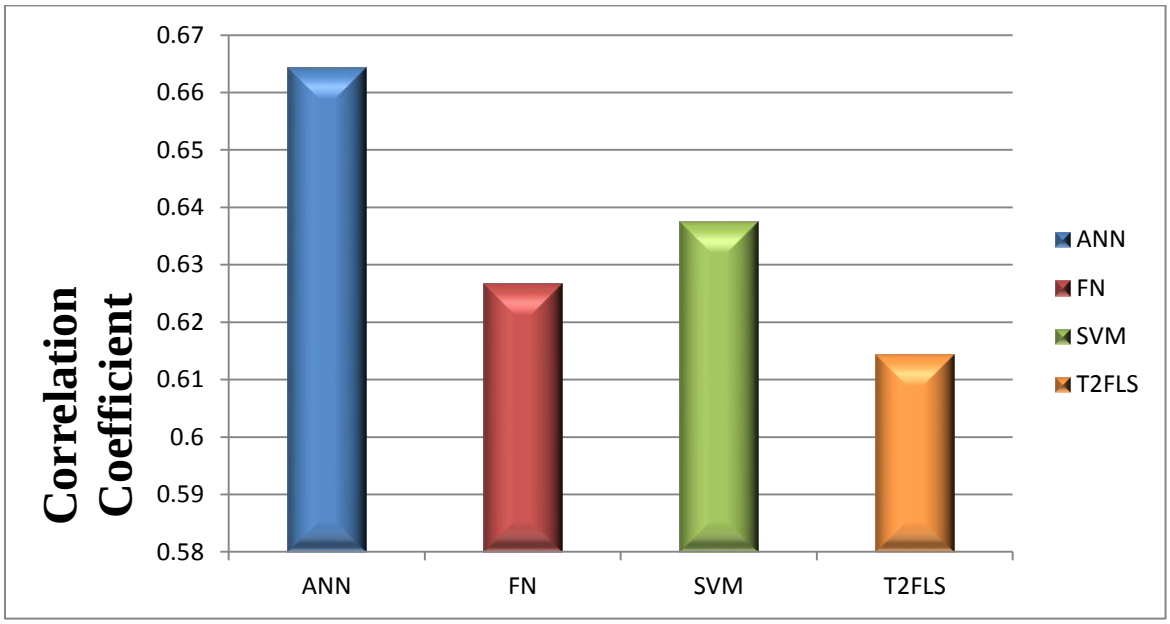


Figure 4-21 Correlation coefficient with Hybrid Feature Selection

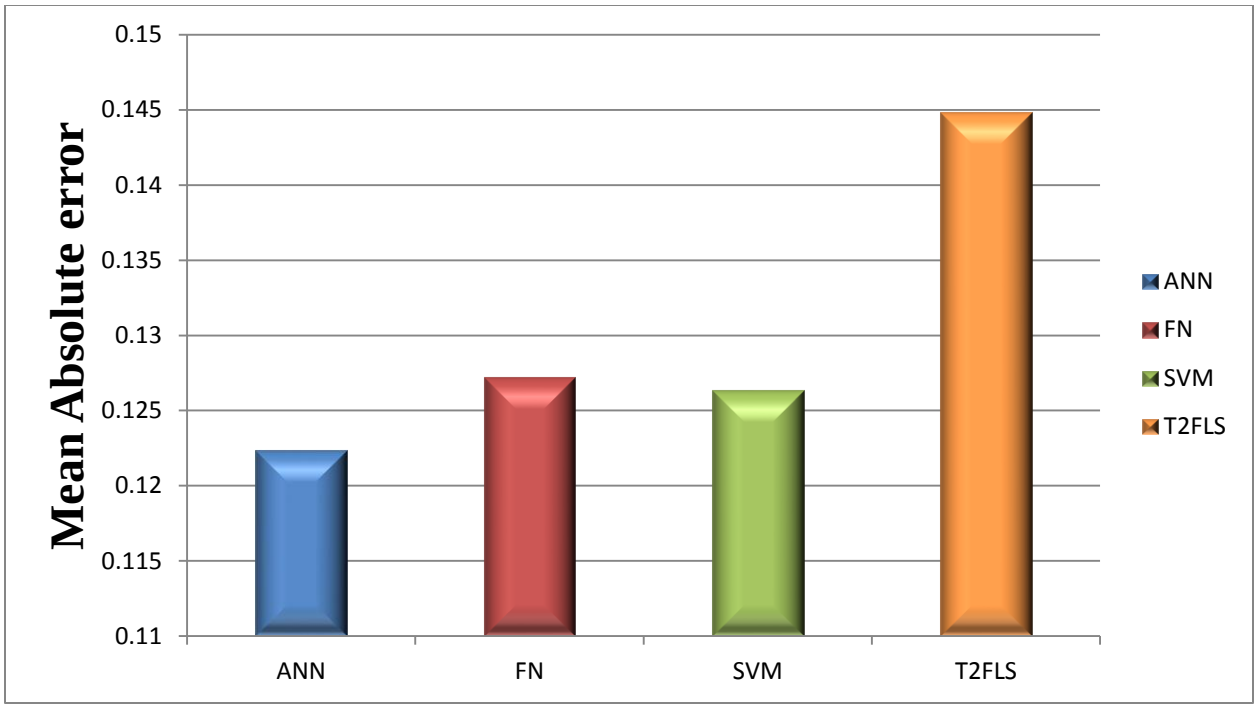


Figure 4-22 Mean absolute error with Hybrid Feature Selection

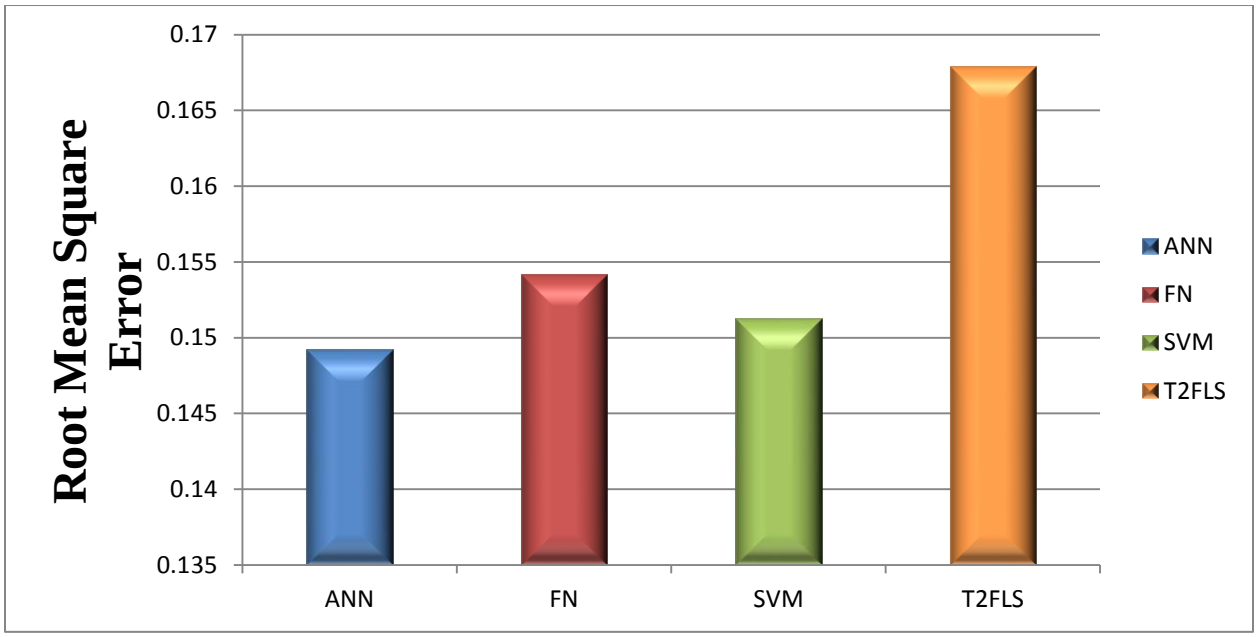


Figure 4-23 Root mean square with Hybrid Feature Selection

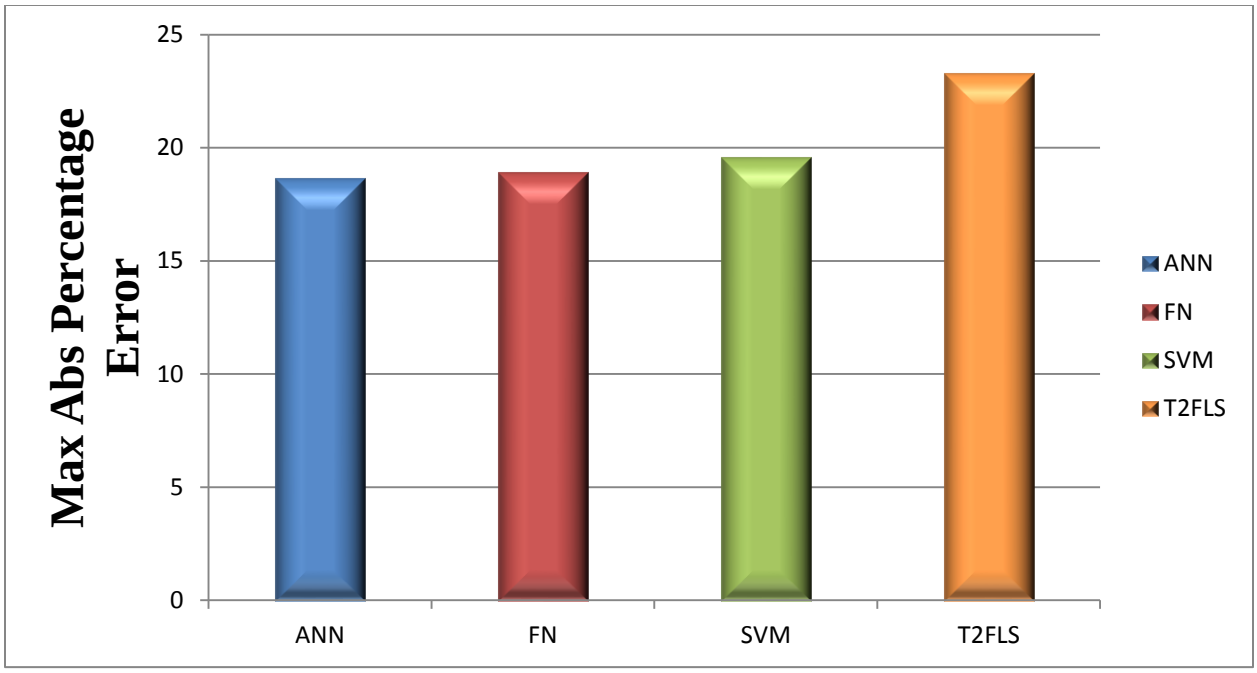


Figure 4-24 Maximum absolute percentage error with Hybrid Feature Selection

CHAPTER 5

Fusion of AI with Archie Formula

5.1 Archie Result as Input

As continuations of our effort of finding the best and the most favorable model that can predict water saturation property in carbonate reservoirs, we examined two approaches in order to achieve this goal. We wanted to help AI as much as possible by providing additional related data as an input to the data set. We calculated water saturation using Archie equation and put the result as an additional input. We followed the same procedure of dividing the data set into three parts, one portion for training and the second portion for validation and finally the last portion for testing. Then we ran all the AI techniques with the new input set.

Figures 5-1 and 5-2 show the results for the T2FLS. With the addition of Archie result as an input we started to see significant improvement in T2FLS prediction capabilities. This is the first time that one can see noticeable positive improvement of the crossplot of T2FLS. There is a slight improvement of agreement between the measured water saturation and predicted water saturation.

The results for SVM are shown in Figures 5-3 and 5-4. Similar to the T2FLS, SVM gained an added value by using Archie results as input. However, in the lower ranges of saturation, the prediction was higher than the measured values. In well#2 predicted water saturation values are in a better agreement with the model than well#1

Figure 5-5 and 5-6 show the results obtained from the functional network model. It seems that FN was also benefiting from this addition of the input dataset and this can be seen in the crossplot. Here too, it can be seen that the model is over predicting at low saturation values. Also, FN performed relatively better in the measured and predicted water saturation versus depth for both wells.

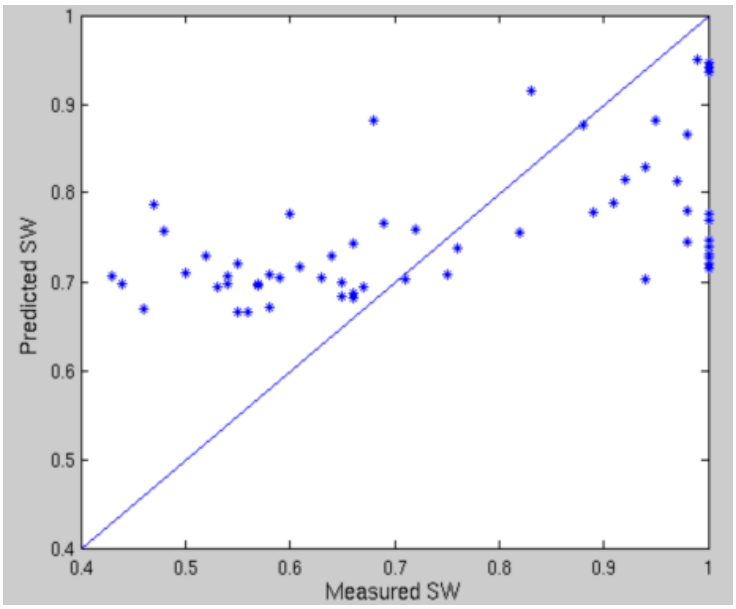


Figure 5-1Crossplot of Measured and Predicted Sw (Testing) using T2FLS

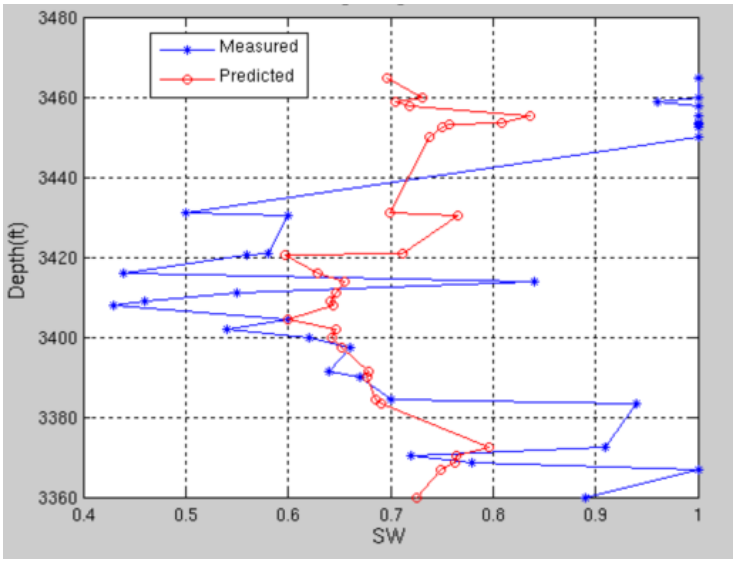


Figure 5-2a Depth vs Sw Testing using T2FLS Model for well1

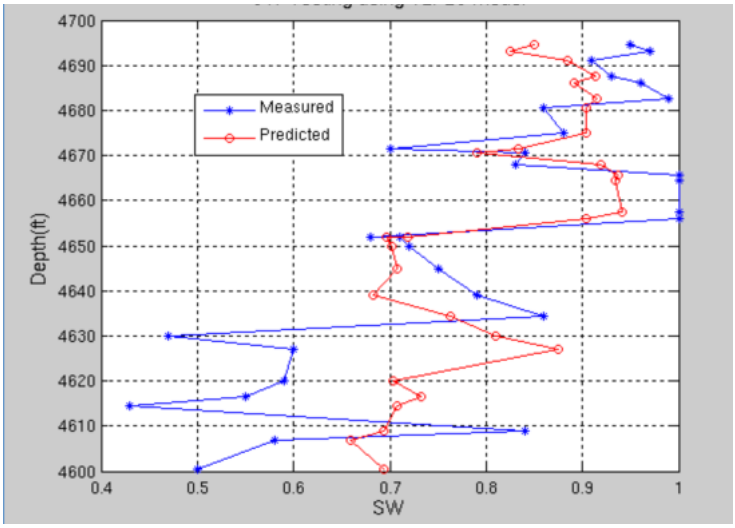


Figure 5-2b Depth vs Sw Testing using T2FLS Model for well2

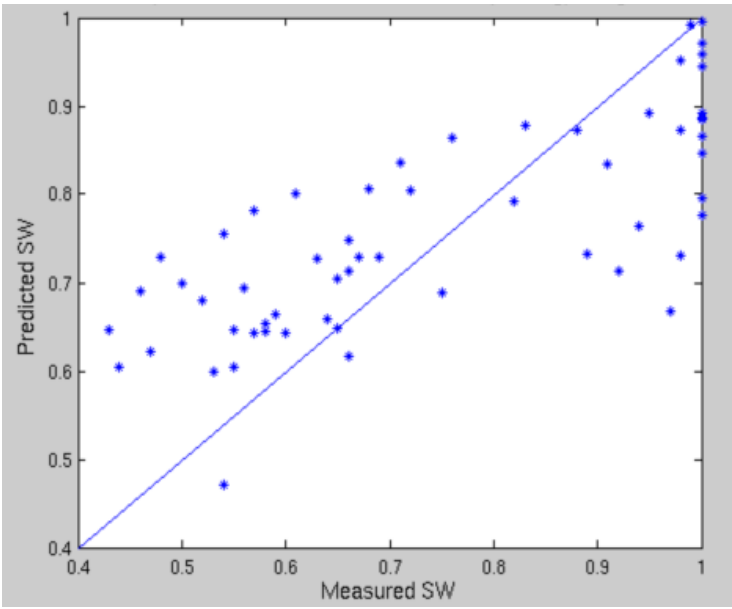


Figure 5-3 Crossplot of Measured and Predicted Sw (Testing) using SVM

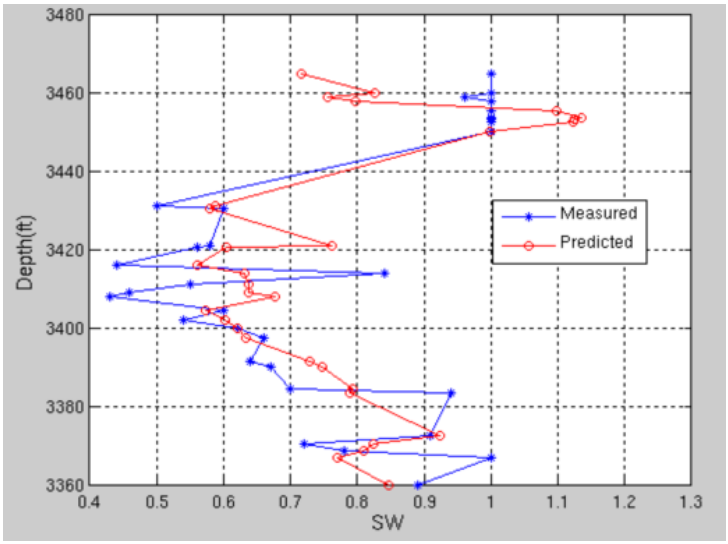


Figure 5-4a Depth vs Sw Testing using SVM Model for well#1

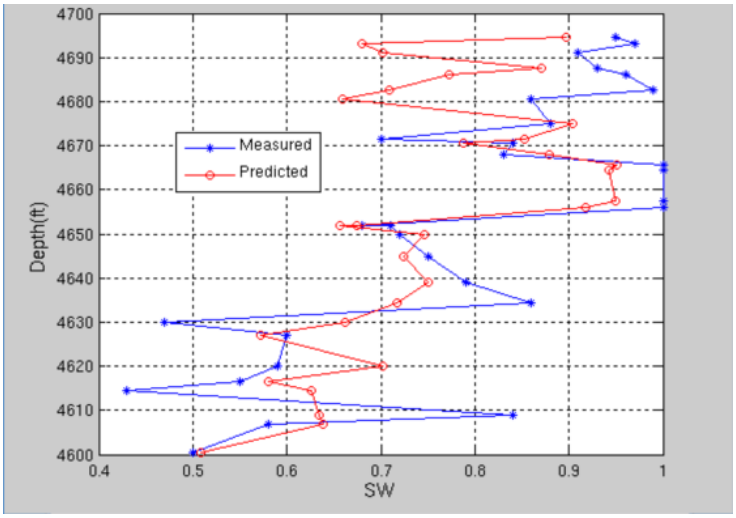


Figure 5-4b Depth vs Sw Testing using SVM Model for well#2

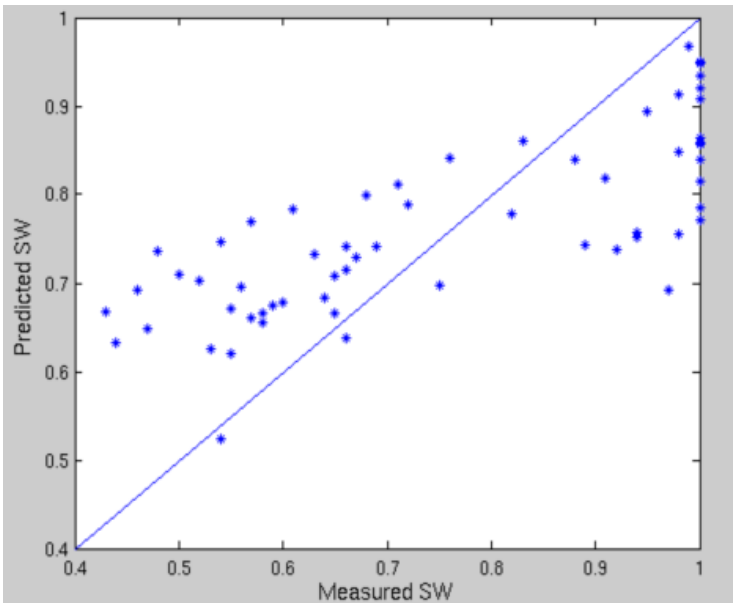


Figure 5-5 Crossplot of Measured and Predicted Sw (Testing) using FN

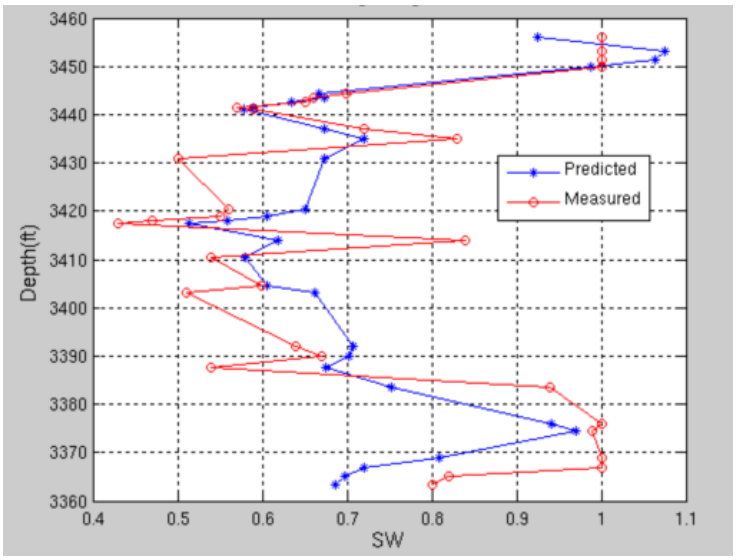


Figure 5-6a Depth vs Sw Testing using FN Model for well#1

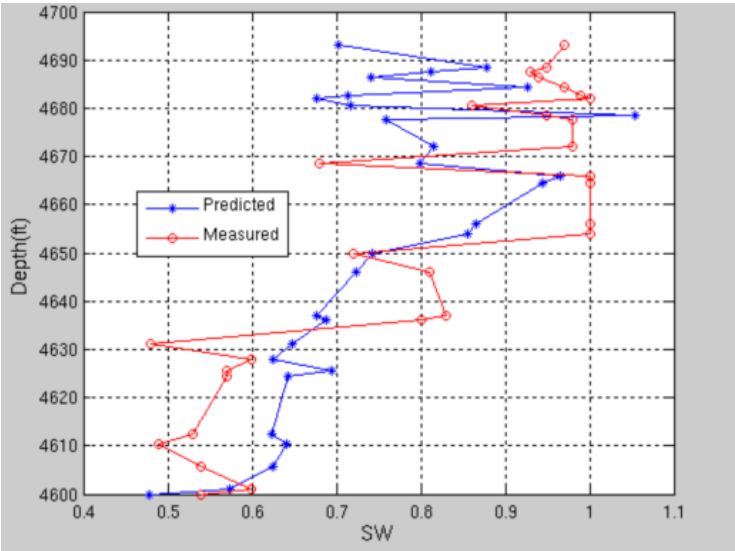


Figure 5-6b Depth vs Sw Testing using FN Model for well#2

Figures 5-7 and 5-8 show the results of the ANN. From the figures, it is clear that ANN did excellent performance compared to all previous AI models. The trend is obvious in both the crossplot as well as in the measured and predicted water saturation versus depth. Both wells show better agreement between the predicted and measured values. Figures 5-9 through 5-12 show the results of the statistical error measurements. ANN scored the highest correlation coefficient and the lowest MAE, RMSE and MAPE. SVM also performed very well in this experiment. As for FN and F2FLS, their results were relatively better compared to previous experiments. Among the four AI techniques, F2SLF has the lowest correlation coefficient and highest MAE, RMSE and MAPE.

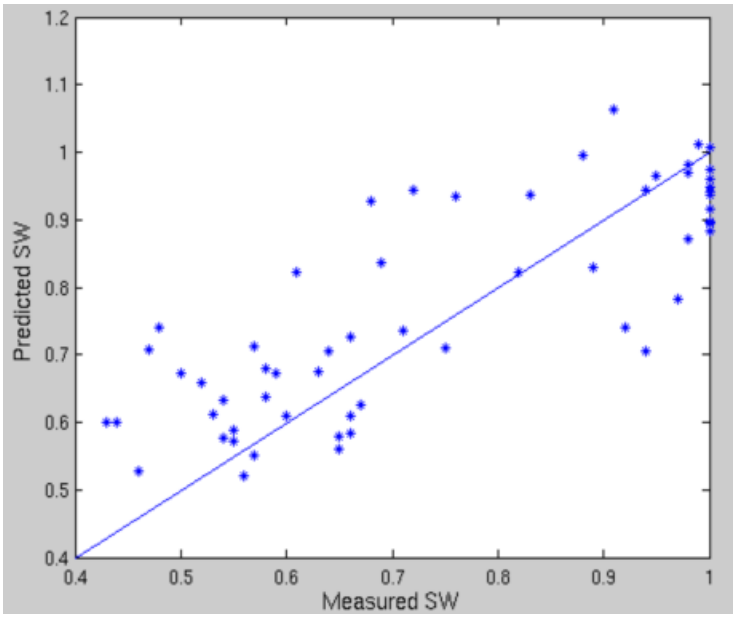


Figure 5-7 Crossplot of Measured and Predicted Sw (Testing) using ANN

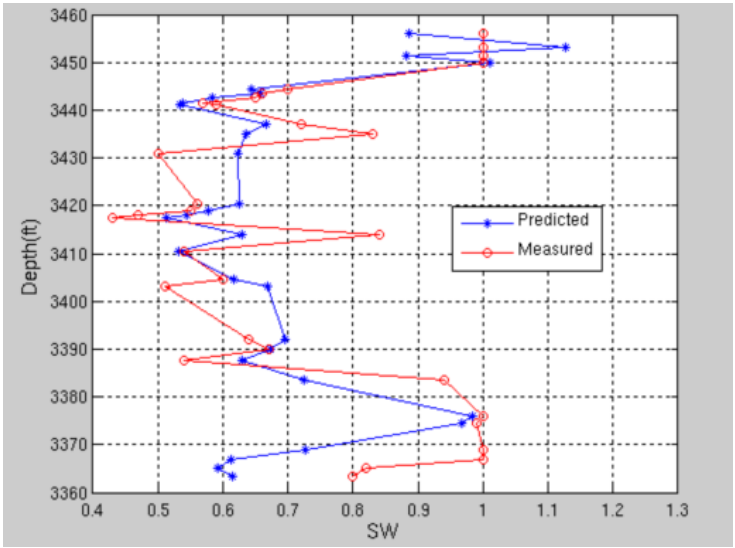


Figure 5-8a Depth vs Sw Testing using ANN Model for well#1

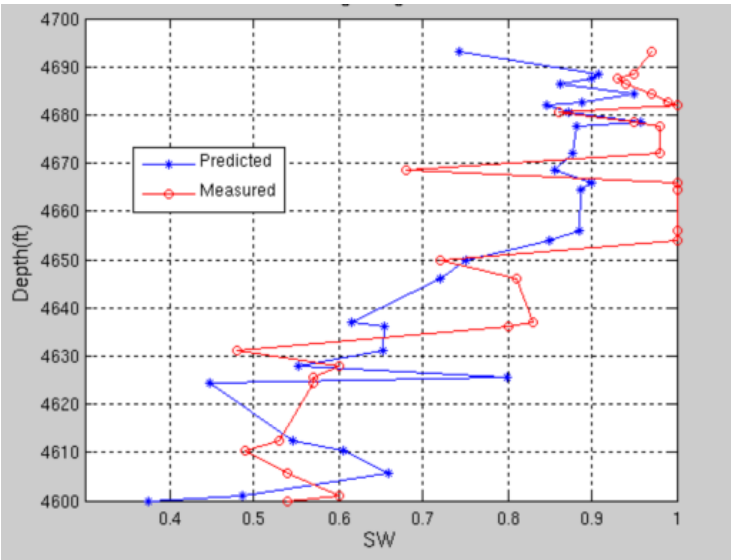


Figure 5-8b Depth vs Sw Testing using ANN Model for well#2

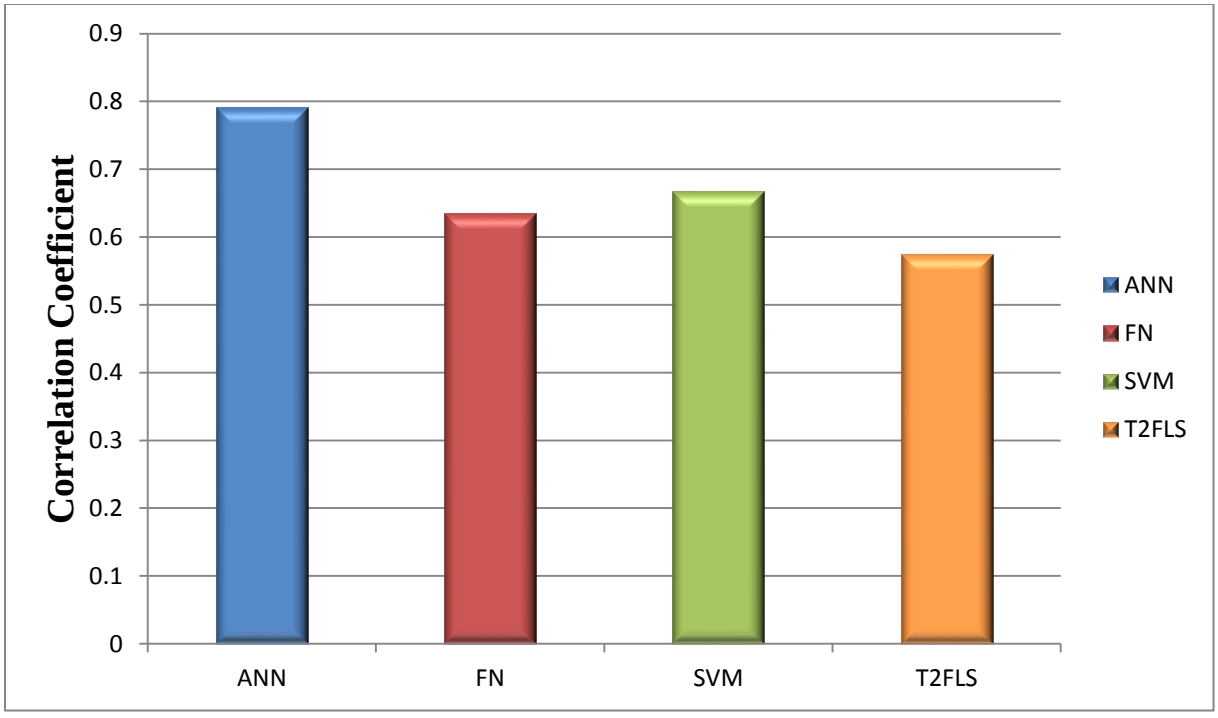


Figure 5-9 Correlation coefficient with Archie Saturation as Input

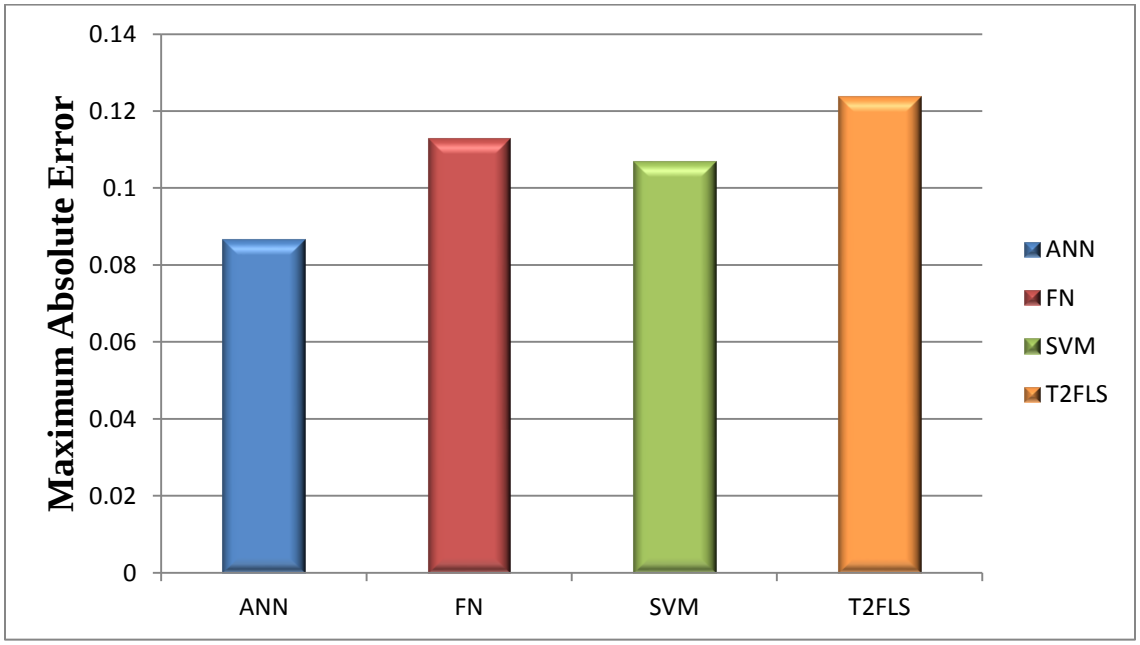


Figure 5-10 Mean absolute error with Archie Saturation as Input.

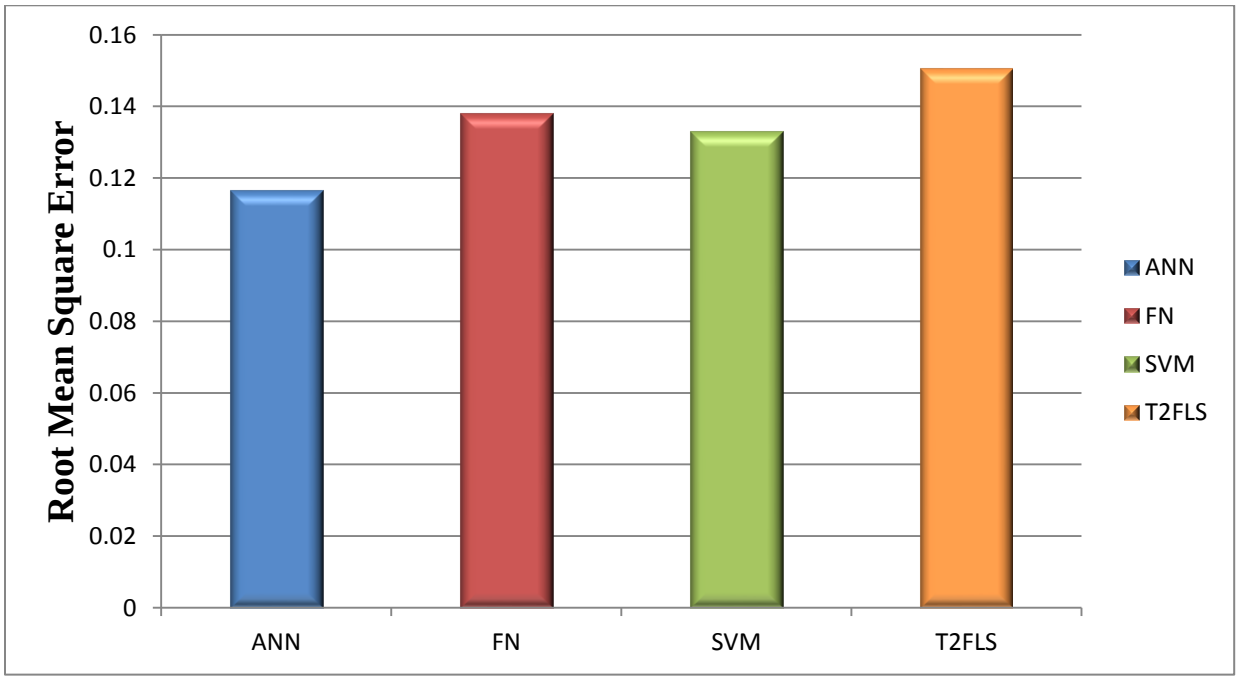


Figure 5-11 Root mean square with Archie Saturation as Input.

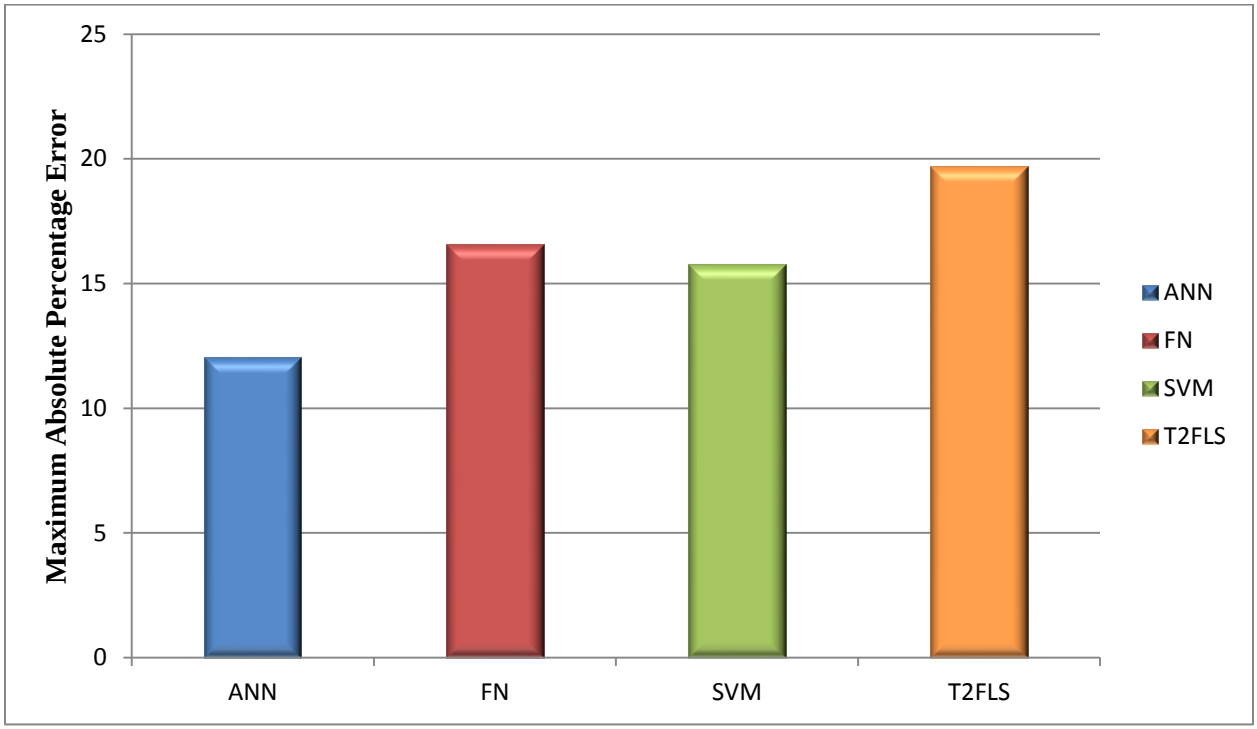


Figure 5-12 Maximum absolute percentage error with Archie Saturation as Input

5.2 Archie Components as Input

The second approach followed here was to break down Archie's equation into two main components. The outcome of each component was added as an additional values for the input data set. The same procedure was followed as in the previous step for the division of data for the training and the testing phases.

$$\left(\frac{1}{\Phi^{m}}\right)^{\frac{1}{n}}, \left(\frac{R_w}{R_t}\right)^{\frac{1}{n}}$$

Figures 5-13 and 5-14 show the results of the T2FLS. With the addition of Archie equation components as additional input, T2FLS continued to improve its prediction capabilities. This can be seen in its crossplot. The improvement is clearly noticeable in the measured and predicted water saturation versus depth in both wells.

The results of SVM are shown in Figures 5-15 and 5-16. In this case also, the crossplots show better agreement between the measured and the predicted water saturation values which indicate a good improvement in the prediction capabilities of SVM. This improvement has also positively contributed to the measured and the predicted water saturation versus depth plots for both wells.

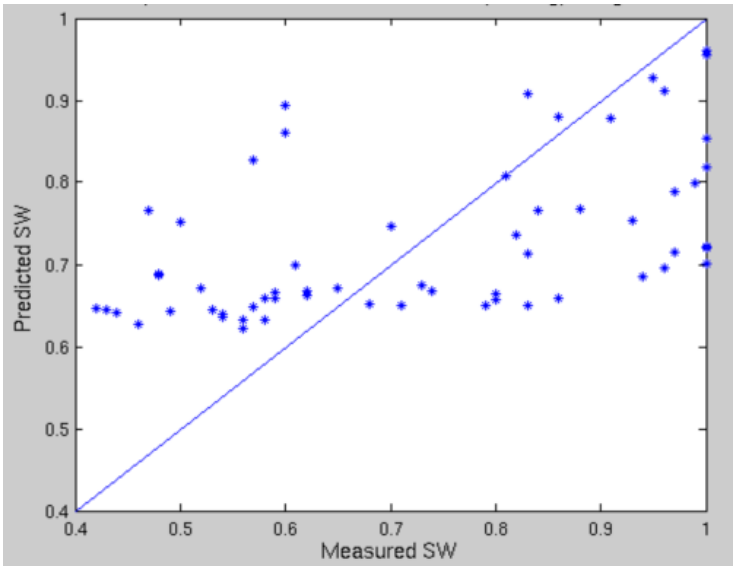


Figure 5-13 Crossplot of Measured and Predicted Sw (Testing) using T2FLS

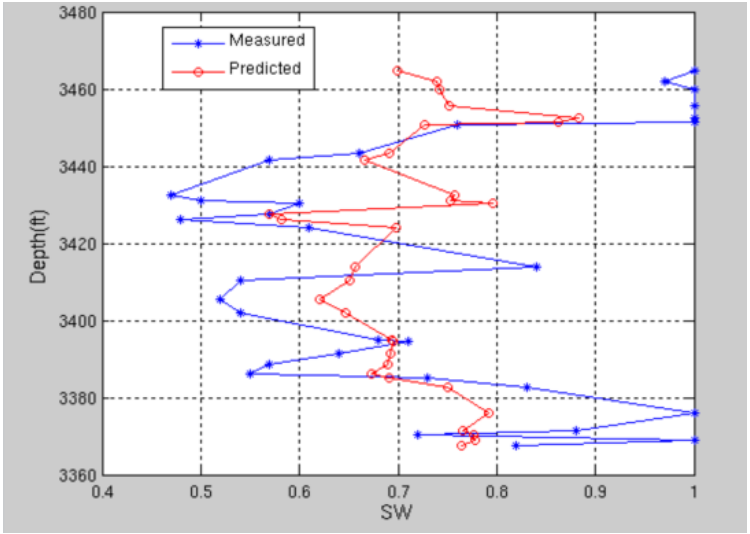


Figure 5-14a Depth vs Sw Testing using T2FLS Model for well#1

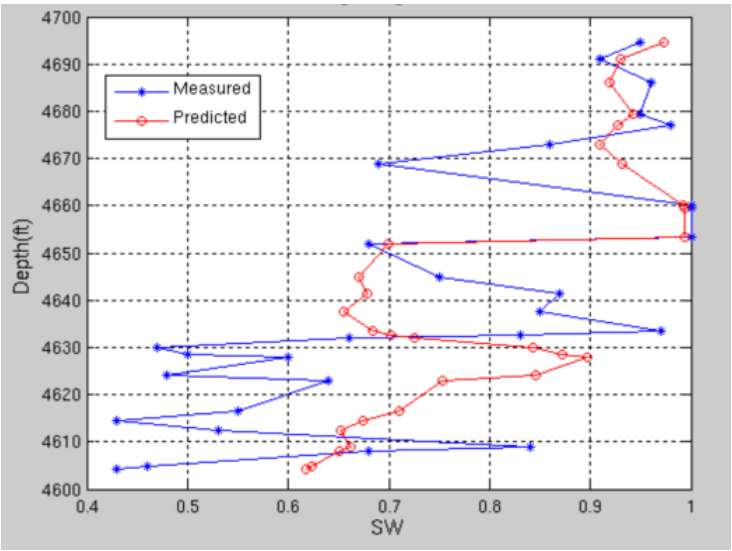


Figure 5-14b Depth vs Sw Testing using T2FLS Model for well#2

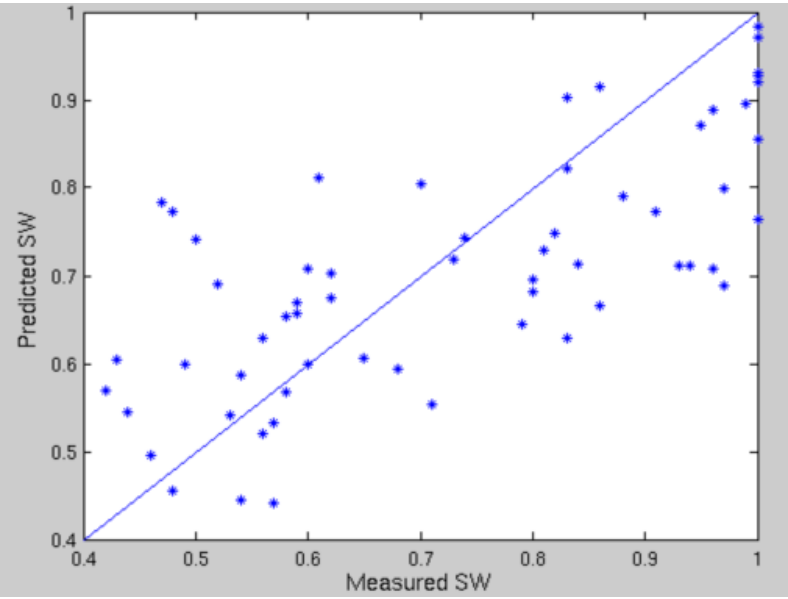


Figure 5-15 Crossplot of Measured and Predicted Sw (Testing) using SVM

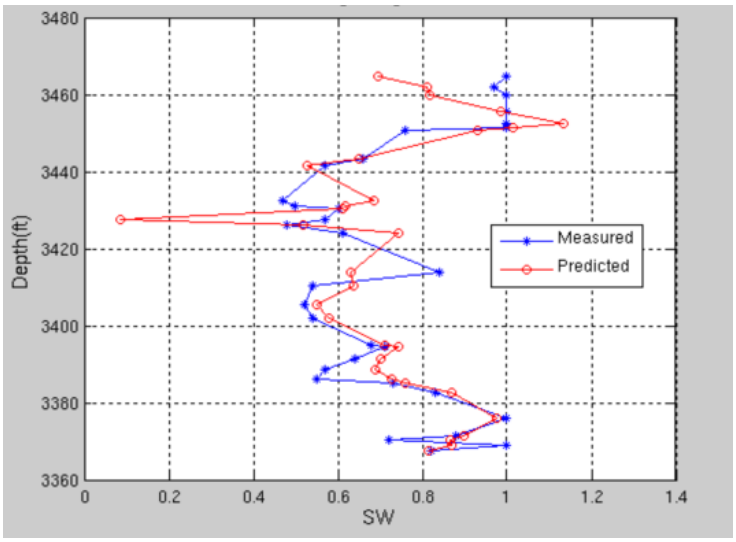


Figure 5-16a Depth vs Sw Testing using SVM Model for well#1

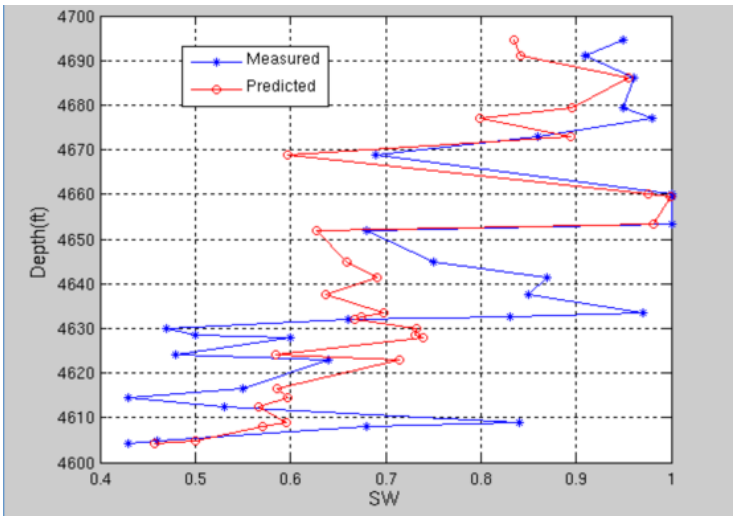


Figure 5-16b Depth vs Sw Testing using SVM Model for well#2

Figures 5-17 and 5-18 show the results of the functional network model. The measured water saturation versus the predicted water saturation crossplot has showed a much better agreement between the predicted values and the measured water saturation values with a few points on the 45 degree line. Saturation values vs depth also show this trend.

The results from ANN model are presented in Figures 5-19 and 5-20. Looking at the measured water saturation versus the predicted water saturation crossplot and the measured water saturation and predicted water saturation versus depth, it can be stated that the performance of ANN has been the best compared to all previous models and approaches. Several points lie either on the 45 degree line within a very close proximity to it.

Figures 5-21 through 5-24 show the results of the statistical error measurements. In this experiment T2FLS, SVM and FN are all have improved their prediction capabilities. Furthermore, ANN has outperformed all other AI techniques with the least statistical error measurement values of MAE, RMSE and MAPE. Among other AI techniques, T2FLS had the least correlation coefficient and the highest values of the other statistical error measurements.

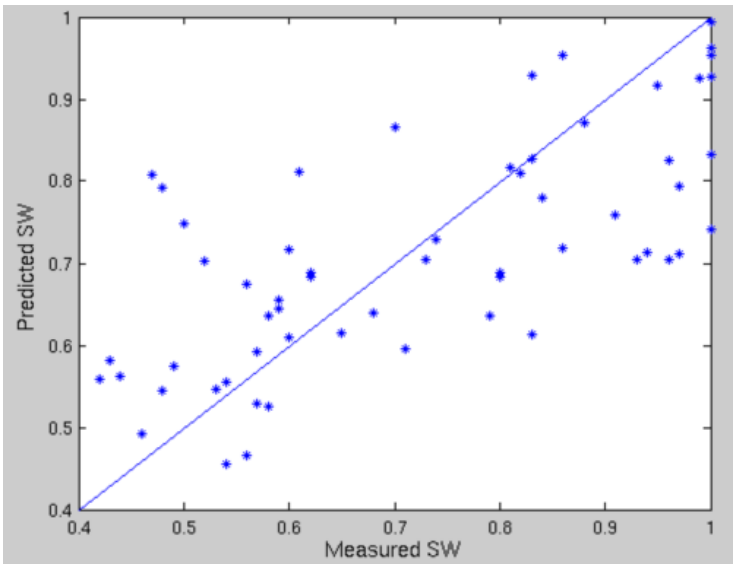


Figure 5-17 Crossplot of Measured and Predicted Sw (Testing) using FN

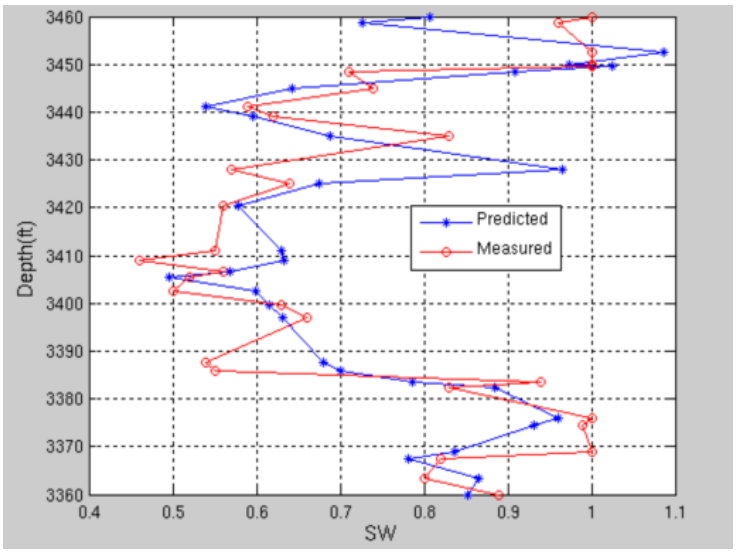


Figure 5-18a Depth vs Sw Testing using FN Model for well#1

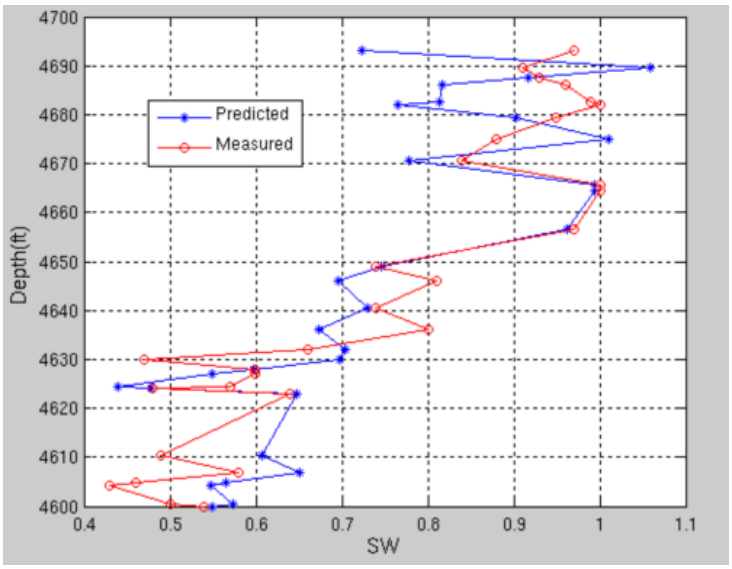


Figure 5-18b Depth vs Sw Testing using FN Model for well#2

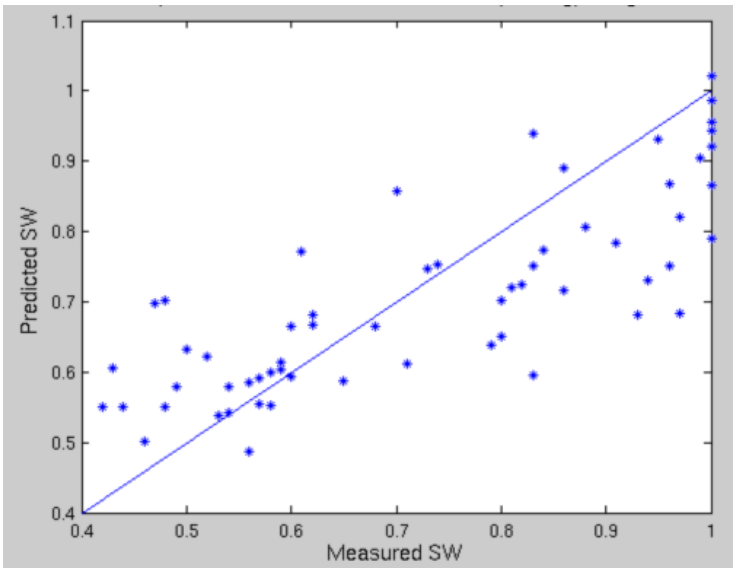


Figure 5-19 Crossplot of Measured and Predicted Sw (Testing) using ANN

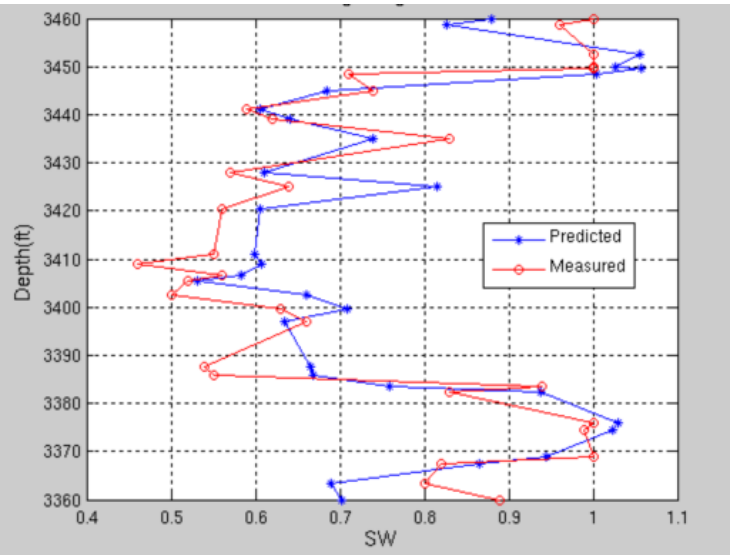


Figure 5-20a Depth vs Sw Testing using ANN Model for well#1

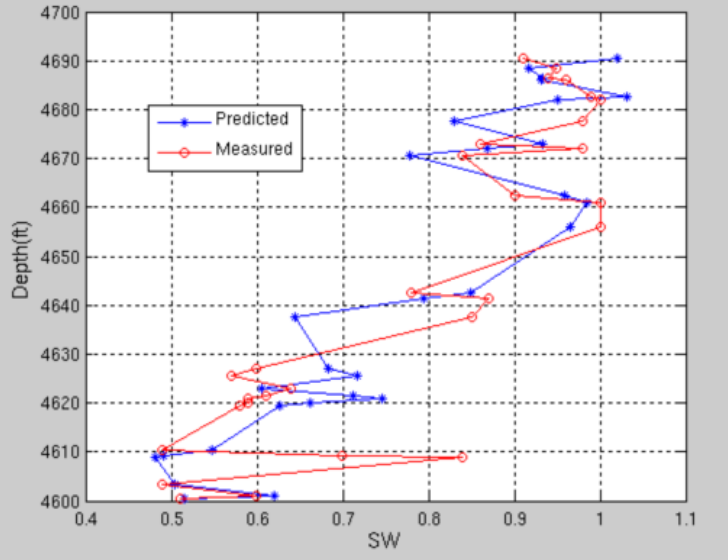


Figure 5-20b Depth vs Sw Testing using ANN Model for well#2

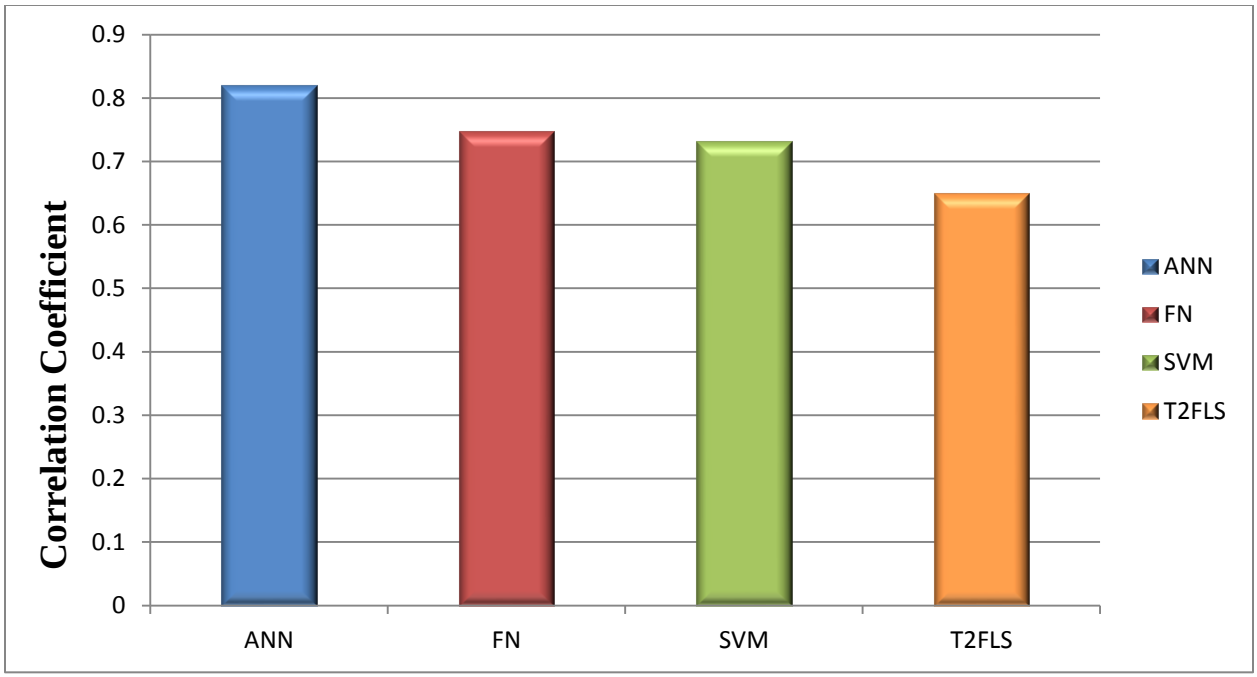


Figure 5-21Correlation coefficient with Components of Archie's equation as Input

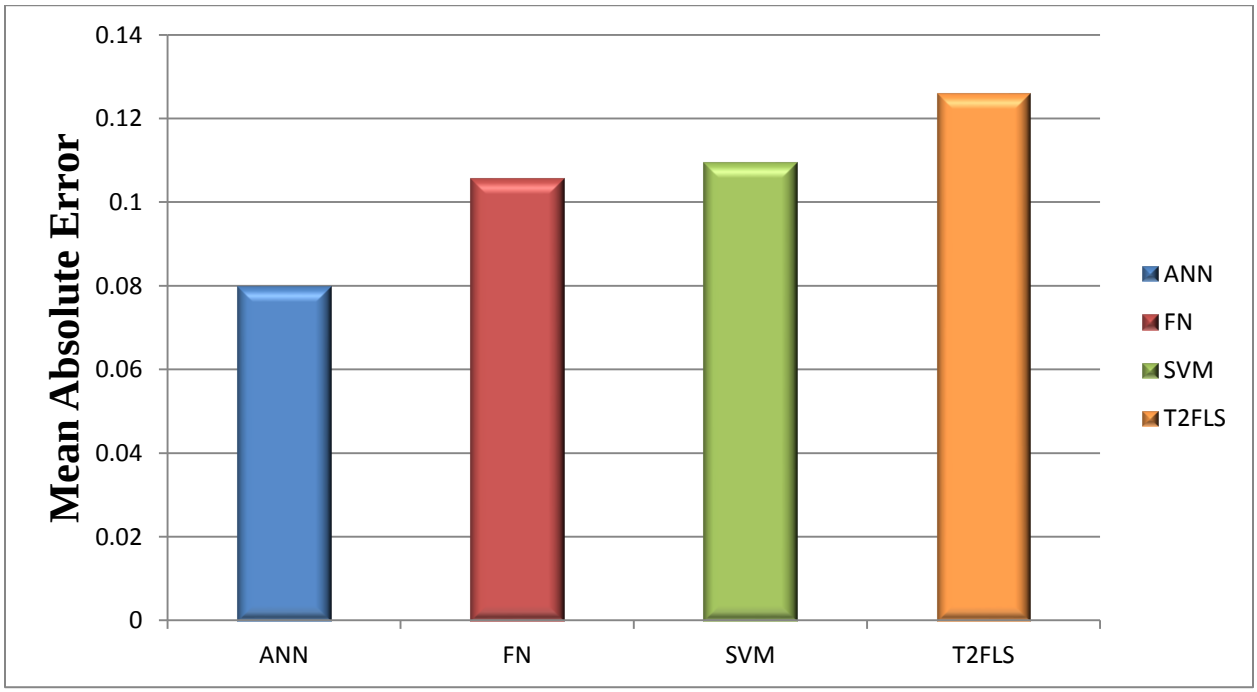


Figure 5-22 Mean absolute error with Components of Archie's equation as Input

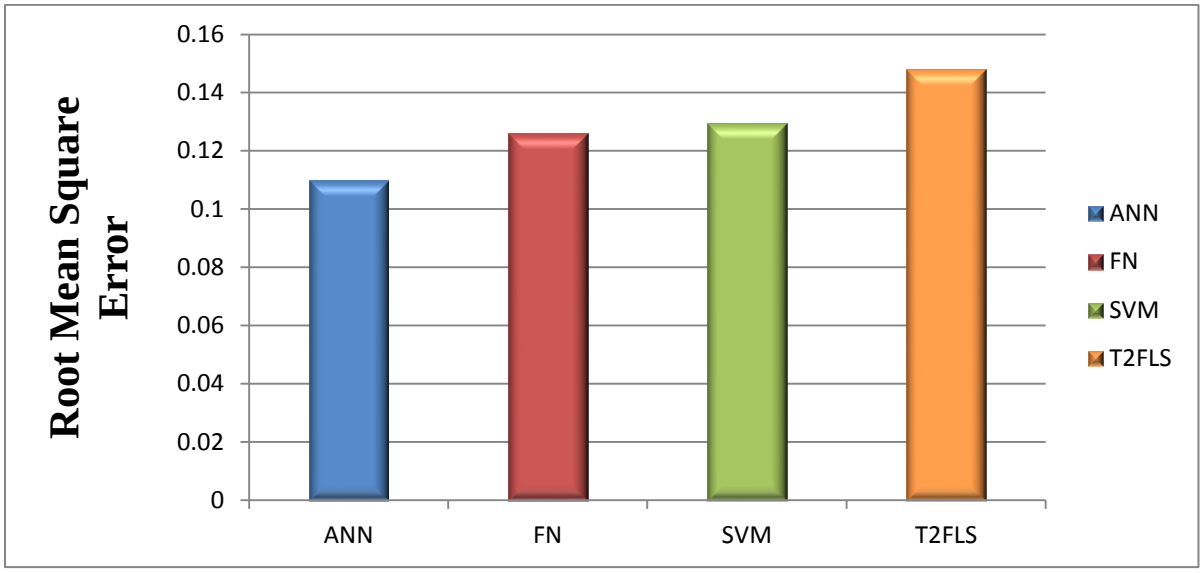


Figure 5-23 Root mean square Error with Components of Archie's equation as Input

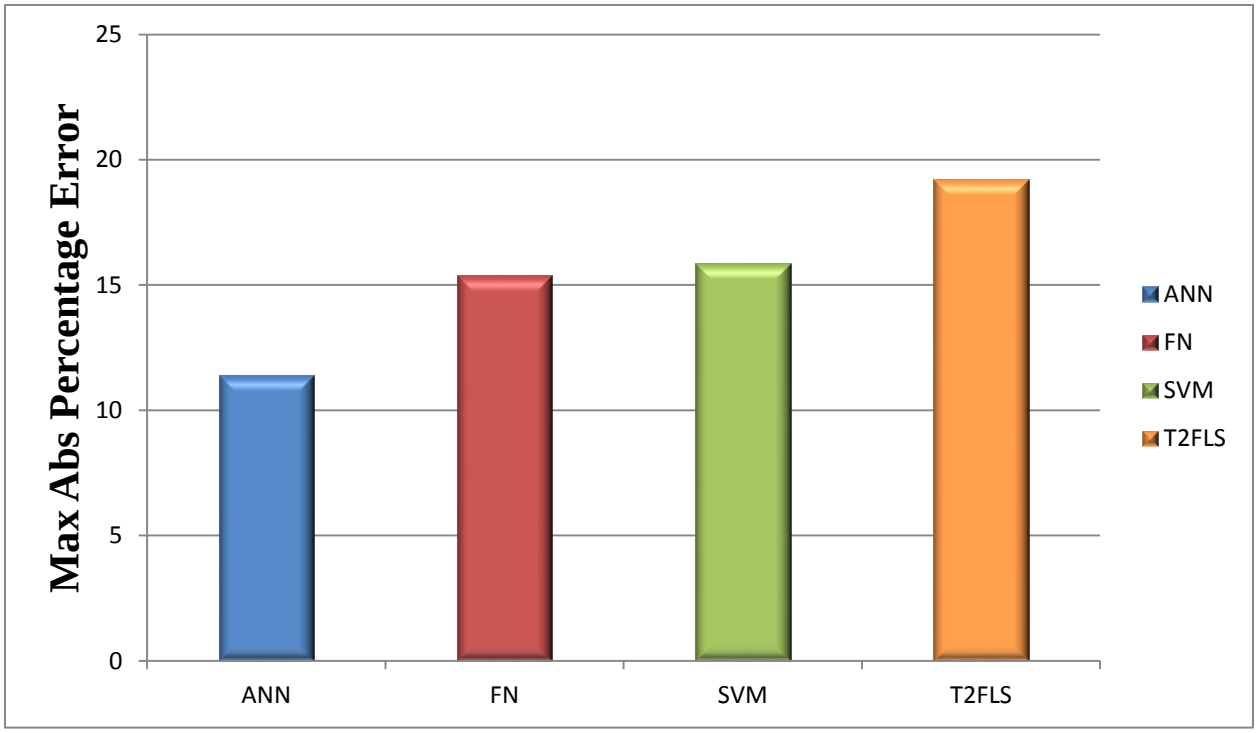


Figure 5-24 Maximum absolute percentage error with Components of Archie's equation as Input

CHAPTER 6

6.1 Conclusions

Looking to the results and discussions offered in the current thesis, pointed summary is listed below:

- The AI models can be used successfully to estimate water saturation from wireline logs and core data in carbonates formations.
- AI techniques can be used as a cost effective alternative to estimate water saturation in carbonate reservoirs.
- Some AI techniques performed better than others in the prediction of saturation which suggests that it is best practice to run several AI techniques to arrive at the best model.
- In this study, ANN outperformed all AI models in addition to Archie model in all scenarios. The predictions from ANN yielded the highest correlation coefficient of (0.82) with respect to measured values.
- Fusion of results from Archie's equation and Archie's equation components have helped all AI models to perform better.

6.2 Recommendations

The following recommendations can be made for extending this study.

- Use parameters from NMR log as additional input data because they describe the pore size distribution of rock in a better way
- Select the right wells in the field location to get core data for training the AI models. This can make an improvement in the prediction capabilities of the models.

Explore additional hybrid optimization techniques to enhance the AI model.

6.3 References

1. Archie, G.E., 1941. The electrical resistivity log as an aid in determining some reservoir characteristics. Transactions of AIME 146, 54–62.
2. Helle, H.B., Bhatt, A., 2002. Fluid saturation from well logs using committee neural networkss. Petroleum Geosciences 8, 109–118
3. Elshafei, M., Hamada G., 2007. Neural Networks Identification of Hydrocarbon Potential of Shaly Sand Reservoirs. Proceeding of SPE Saudi Arabia Technical Symposium, Dhahran 7-8 May 2007
4. Ali, J.K., 1994. Neural networkss: a new tool for petroleum industry? Proceedings of the SPE European Petroleum Computer Conference, UK. Paper SPE, 27561, p. 15
5. Mohaghegh, S., Arefi, R., Bilgesu, I., Ameri, S., Rose, D., 1994. Design and development of an artificial neural networks for estimation of formation permeability. Proceeding of SPE Petroleum Computer Conference, Dallas Paper SPE, 28237, p. 4.
6. Haykin, S., 1998. Neural networkss, a comprehensive foundation. Prentice Hall PTR, USA
7. Olson, T.M., 1998. Porosity and permeability prediction in low- permeability gas reservoirs from well logs using neural networkss. Proceeding of the SPE Rocky Mountain Regional Symposium and Exhibition, Colorado. Paper SPE, 39964, 10
8. Bazzaz, W. H.; Y. W. Mehanna; A. Gupta; 2007. Permeability Modeling Using Neural-Networks Approach for Complex Mauddud-Burgan Carbonate Reservoir. Proceeding of the 15th SPE Middle East Oil & Gas Show and Conference held in Bahrain, 11-14 March 2007
9. Anifowose, F. A., Abdulraheem, A., Al-Shuhail, A. A., & Schmitt, D. P. (2013, March 10). Improved Permeability Prediction From Seismic and Log Data using Artificial Intelligence Techniques. Society of Petroleum Engineers. doi:10.2118/164465-MS
10. Gevrey, M., Dimopoulos, I., Sovan, L., 2003. Review and comparison of methods to study the contribution of variables in artificial neural networks models. Ecological Modeling 160, 249–264
11. Al-Bulushi, N., King, P., Blunt, M., Kraaijveld, M., 2010. Generating a capillary saturation-height function to predict hydrocarbon saturation using artificial neural networkss. Petroleum Geoscience, Vol. 16 2010, pp. 77–85
12. Shamsi, A. Talia, Talabani, Soran, Vaziri, H.H., Islam, M.R. 2001. In-Depth Investigation of the Validity of the Archie Equation in Carbonate Rocks. SPE POS Oklahoma City, Oklahoma, 24–27 March 2001

13. Masoudi, R., Halim, M., Karkooti, H. 2011. On the concept of water saturation determination and modeling in carbonate reservoirs. IPTC in Bangkok, Thailand 7-9 February 2012.

14. Richardson, J. G., & Holstein, E. D. (1994, January 1). Comparison Of Water Saturations From Capillary Pressure Measurements With Oil-Base-Mud Core Data Ivishak (Sadlerochit) Reservoir, Prudhoe Bay Field. Society of Petroleum Engineers. doi:10.2118/28593-MS

15. Bust, V. K., Oletu, J. U., & Worthington, P. F. (2009, January 1). The Challenges for Carbonate Petrophysics in Petroleum Resource Estimation. International Petroleum Technology Conference. doi:10.2523/13772-MS

16. Al-Bulushi, N., King, P., Blunt, M., Kraaijveld, M., 2009. Development of artificial neural networks models for predicting water saturation and fluid distribution. Journal of Petroleum Science and Engineering 68 (2009) 197–208.

17. Mohammad, A. Ibrahim, Potter K. David. 2004. Prediction of residual water saturation using genetically focused neural nets. SPE Asia Pacific Oil and Gas Conference and Exhibition held in Perth, Australia, 18-20 October 2004.

18. Anifowose, Fatai A., Abdulraheem, A., Al-Shuhail, A. A., & Schmitt, D. P. (2013, March 10). Improved Permeability Prediction From Seismic and Log Data using Artificial Intelligence Techniques. Society of Petroleum Engineers. doi:10.2118/164463-MS

19. Anifowose, Fatai Adesina, & Abdulraheem, A. (2010, January 1). Prediction of Porosity and Permeability of Oil and Gas Reservoirs using Hybrid Computational Intelligence Models. Society of Petroleum Engineers. doi:10.2118/126649-MS

20. Anifowose, Fatai Adesina, Ewenla, A. O., & Eludiora, S. I. (2011, January 1). Prediction of Oil and Gas Reservoir Properties using Support Vector Machines. International Petroleum Technology Conference. doi:10.2523/14514-MS

21. Fleury, M., Al-Nayadi, K., Boyd, D., 2005. Water Saturation from NMR, Resistivity and Oil Base Core in a Heterogeneous Middle East Carbonate Reservoir. Proceeding of the SPWLA 46th Annual Logging Symposium held in New Orleans, Louisiana, United States, June 26-29, 2005.

22. Bello, O., & Asafa, T. (2014, April 1). A Functional Networks Softsensor for Flowing Bottomhole Pressures and Temperatures in Multiphase Production Wells. Society of Petroleum Engineers. doi:10.2118/167881-MS

6.4 Appendix

Program Listing

A.1 Program to calculate the models Parameters

```
%This single M-file runs 4 AI Techniques: ANN, FN, SVM and Type-2 Fuzzy.
% The purpose is to AUTOMATICALLY determine which stratification option is the best.
% From 99 to 1 with increment of 1.

clear all; close all; clc;
warning off all;
%format short;

disp('------------------------------------------------------------------');
disp('WELCOME to the PARAMETRIC STUDY TEST PROGRAM');
disp('IMPLEMENTED ON ANN, SVM and T2FLS');
disp('------------------------------------------------------------------');
disp(' ');
disp('Loading Data ... Press any key to continue');
disp(' ');
%pause;

%Reading Data
X = xlsread('C:\Users\User 1\MyMATLAB\data\input_target_data.xlsx','b3:g204');

%Reading Target
D = xlsread('C:\Users\User 1\MyMATLAB\data\input_target_data.xlsx','h3:h204');

data=[X,D];

disp(' ');
disp('Data Successfully Loaded');
disp(' ');

[n,p]=size(X);

fprintf('Number of observations: = %5.0f \n',n),
fprintf('Number of features: = %5.0f \n',p),
disp(' ');

% ANN parameters
%numHiddenNeurons = 15; %%This is the optimal (Fatai) %40; %10;  % Adjust as desired
net.trainParam.epochs = 1000;

%SVM parameters
% C =  450; %100; %450; %1000; %1500; %2500; %3500; %4500;
```

```
% lambda = 1e-9; 1e-7;
% epsilon = 0.6; 0.2; %.05;
kerneloption = 0.30;
kernel='none' ;'poly';
verbose=2;

%FunNet parameters
k=1;

%T2F parameters
% alpha= 0.1; %0.2; %1.0; % 0.15; %1.0; %0.1; %1.5;   %0.5; %0.01;   %0.0001;%0.001;
%c1;
% alpha4=alpha;

%Stratification Initialization in a Loop
%for i=0.9:0.1:-0.1
% disp('-----------------------------------------------------');
% num_of_runs = input('Enter desired number of runs (5 recommended due to time):> ');
% disp('-----------------------------------------------------');

% disp('-----------------------------------------------------');
% choice = input('Enter desired Stratification (Must be a fraction):> ');
% disp('-----------------------------------------------------');

%for choice=96:-1:92

%Stratification initialization
choice = 0.7;

ANNcctr=[];
ANNccts=[];
SVMcctr=[];
SVMccts=[];
T2Fcctr=[];
T2Fccts=[];

    disp(' ');
    disp('ANN Section Started');
    disp(' ');

    index=1:n;index=index';

s0=stratif(n,ceil(choice*n));
X_tr = X(s0,:); D_tr = D(s0,:); data_tr=data(s0,:); %depth_tr=depth(s0,:);
index0=index; index0(s0)=[];
X_ts = X(index0,:); D_ts = D(index0,:); data_ts = data(index0,:); %depth_ts =
depth(index0,:);

[n_tr,p]=size(X_tr);

for numHiddenNeurons = 1:10:100

% index=1:n;index=index';
%
% s0=stratif(n,ceil(choice*n));
% X_tr = X(s0,:); D_tr = D(s0,:); data_tr=data(s0,:); %depth_tr=depth(s0,:);
% index0=index; index0(s0)=[];
%  X_ts = X(index0,:); D_ts = D(index0,:); data_ts = data(index0,:); %depth_ts =
depth(index0,:);
%
%     %Stratification Process finished
%
%     [n_tr,p]=size(X_tr);
```

```
    %Transposing the input and output for ANN
    X_tr1=X_tr';
    D_tr1=D_tr';

    % ANN Starts here
    tr_starttime = cputime;                % Set time clock

    % Create a FFBP ANN
net = feedforwardnet(numHiddenNeurons);
net.trainParam.epochs = 50;

net = train(net,X_tr1,D_tr1);

%view(net)


%Evaluate with Training data
outputtr = net(X_tr1);

Timetr = cputime - tr_starttime;

    % ========================================================
    %Corellation between training output and values predicted with training data
    % ========================================================
    varx=sum(outputtr.^2)/n_tr - (sum(outputtr)^2)/(n_tr^2);
    vary=sum(D_tr1.^2)/n_tr - (sum(D_tr1)^2)/(n_tr^2);
    covxy=sum(outputtr.*D_tr1)/n_tr  - (sum(outputtr)*sum(D_tr1))/(n_tr^2);
    cctr=covxy/(sqrt(varx * vary));

    % ========================================================
    % Calculating Root mean Square Error for Training
    % ========================================================
    RMSEtr = errperf(D_tr1,outputtr,'rmse');
    MAEtr = errperf(D_tr1,outputtr,'mae');
    MAPEtr = errperf(D_tr1,outputtr,'mape');
    Er = D_tr1-outputtr;
    MIN = min(Er);
    MAX = max(Er);


    %Testing begins here
    [n_ts,p]=size(X_ts);

    %Also, Transposing the input and output for ANN
    X_ts1=X_ts';
    D_ts1=D_ts';

    ts_starttime = cputime;                % Set time clock

    %Really Testing with Unseen Data
    outputts = net(X_ts1);

    % ========================================================
    % Get computation time for training
    % ========================================================
    Timets = cputime - ts_starttime;

    % ========================================================
    %Corellation between training output and values predicted with Testing data
    % ========================================================
    varx=sum(outputts.^2)/n_ts - (sum(outputts)^2)/(n_ts^2);
    vary=sum(D_ts1.^2)/n_ts - (sum(D_ts1)^2)/(n_ts^2);
```

```
    covxy=sum(outputts.*D_ts1)/n_ts  - (sum(outputts)*sum(D_ts1))/(n_ts^2);
    ccts=covxy/(sqrt(varx * vary));

    % ============================================================
    % Calculating Root mean Square Error for Training
    % ============================================================
    RMSEts = errperf(D_ts1,outputts,'rmse');
    MAEts = errperf(D_ts1,outputts,'mae');
    MAPEts = errperf(D_ts1,outputts,'mape');
    Er = D_ts1-outputts;
    MIN = min(Er);
    MAX = max(Er);



    ANNcctr = [ANNcctr cctr];
    ANNccts = [ANNccts ccts];


end

% ============================================================
% Plotting results for training and testing for the ANN
% ============================================================
figure('name','Optimal    Training/Testing    Number    of    Hidden    Neurons    for
ANN','NumberTitle','off')
q=1:length(ANNcctr);

plot(q,ANNcctr,'b+-');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(q,ANNccts,'r*-');               %plot(tr_Y(:,s),
xlabel(['Number of Hidden Neurons'],'FontSize',12);
%ylabel(['Porosity'],'FontSize',12);
ylabel(['R-Square'],'FontSize',12);
grid on;
title('Optimal Number of Hidden Neurons for ANN','FontSize',12)
%title('Optimal Number of Hidden Neurons for ANN (Permeability Data 1)','FontSize',12)
legend('Training R-Square','Testing R-Square')

    % SVM Starts Here
    disp(' ');
    disp('SVM Started');
    disp(' ');

    lambda = 1e-7;
epsilon = 0.2; %.05;
for C = 10:1000:10000

%      index=1:n;index=index';
%
%     s0=stratif(n,ceil(choice*n));
%     X_tr = X(s0,:); D_tr = D(s0,:); data_tr=data(s0,:); %depth_tr=depth(s0,:);
%     index0=index; index0(s0)=[];
%      X_ts = X(index0,:); D_ts = D(index0,:); data_ts = data(index0,:); %depth_ts =
depth(index0,:);


    % Identifying the input and output for training
    x=X_tr;
    ytr=D_tr;

    % Initializing the CPU Start Time
    starttime = cputime;
```

```
    %Training SVM
    [xsup,ysup,w,w0] = svmreg(x,ytr,C,epsilon,kernel,kerneloption,lambda,verbose);

    %Calculating the time for training (SVM)
    Timetr = cputime - starttime;

    % Testing the Training
    ypredtr = svmval(x,xsup,w,w0,kernel,kerneloption);

    %Calculating RMSE for the training set (SVM)
    RMSEtr = errperf(ytr,ypredtr,'rmse');
    MAEtr = errperf(ytr,ypredtr,'mae');
    MAPEtr = errperf(ytr,ypredtr,'mape');
    Er = ytr-ypredtr;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the training set (SVM)
    varx=sum(ypredtr.^2)/n_tr - (sum(ypredtr)^2)/(n_tr^2);
    vary=sum(ytr.^2)/n_tr - (sum(ytr)^2)/(n_tr^2);
    covxy=sum(ypredtr.*ytr)/n_tr  - (sum(ypredtr)*sum(ytr))/(n_tr^2);
    cctr=covxy/(sqrt(varx * vary));

    % Identifying the test data
    x=X_ts;
    yts=D_ts;

    % Initializing the CPU Start Time
    starttime = cputime;

    % Testing with unseen Data
    ypredts = svmval(x,xsup,w,w0,kernel,kerneloption);

    %Calculating the time for testing (SVM)
    Timets = cputime - starttime;

    %Calculating RMSE for the testing set (SVM)
    RMSEts = errperf(yts,ypredts,'rmse');
    MAEts = errperf(yts,ypredts,'mae');
    MAPEts = errperf(yts,ypredts,'mape');
    Er = yts-ypredts;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the testing set (SVM)
    varx=sum(ypredts.^2)/n_ts - (sum(ypredts)^2)/(n_ts^2);
    vary=sum(yts.^2)/n_ts - (sum(yts)^2)/(n_ts^2);
    covxy=sum(ypredts.*yts)/n_ts  - (sum(ypredts)*sum(yts))/(n_ts^2);
    ccts=covxy/(sqrt(varx * vary));


    SVMcctr = [SVMcctr cctr];
    SVMccts = [SVMccts ccts];


end


% ========================================================
% Plotting results for training and testing for the ANN
% ========================================================
figure('name','Optimal Value of C for SVM','NumberTitle','off')
```

```
q=1:length(SVMcctr);
plot(q,SVMcctr,'b+-');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(q,SVMccts,'r*-');               %plot(tr_Y(:,s),
xlabel(['Values of C'],'FontSize',12);
ylabel(['R-Square'],'FontSize',12);
grid on;
title('Optimal Value of C for SVM','FontSize',12)
%title('Optimal Number of Hidden Neurons for ANN (Permeability Data 1)','FontSize',12)
legend('Training R-Square','Testing R-Square')

    % Type-2 Fuzzy Starts Here
    disp(' ');
    disp('T2-Fuzzy Started');
    disp(' ');

for alpha = 0:10
    alpha4=alpha;

%      index=1:n;index=index';
%
%     s0=stratif(n,ceil(choice*n));
%     X_tr = X(s0,:); D_tr = D(s0,:); data_tr=data(s0,:); %depth_tr=depth(s0,:);
%     index0=index; index0(s0)=[];
%      X_ts = X(index0,:); D_ts = D(index0,:); data_ts = data(index0,:); %depth_ts =
depth(index0,:);

    % Initializing Variables and Constants
    M1=X_tr;M2=X_tr;c1=D_tr;
    c2=c1;

    for f=1:size(X_tr,1)
        for j=1:size(X_tr,2)
            sigma(f,j)=(1.4272 -M1(f,j))/2; % This is option [3]
        end
    end

    sn1 = std(X_tr);
    sn2=sn1;

    % Initializing CPU Start Time
    starttime = cputime;

    %Training T2F


    %Cummulating the time for training (T2F)
    Timetr = cputime - starttime;

    % Testing the Training
    [R1,R2,R]=sfls_type2(X_tr,M1,M2,sigma,c1,c2);

    R=R';

    %Calculating RMSE for the training set (T2F)
    RMSEtr = errperf(D_tr,R,'rmse');
    MAEtr = errperf(D_tr,R,'mae');
    MAPEtr = errperf(D_tr,R,'mape');
    Er = D_tr-R;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the training set (T2F)
```

```
    varx=sum(R.^2)/n_tr - (sum(R)^2)/(n_tr^2);
    vary=sum(D_tr.^2)/n_tr - (sum(D_tr)^2)/(n_tr^2);
    covxy=sum(R.*D_tr)/n_tr  - (sum(R)*sum(D_tr))/(n_tr^2);
    cctr=covxy/(sqrt(varx * vary));


    % Initializing CPU Start Time
    starttime = cputime;

    % Testing with unseen Data
    [R1,R2,R]=sfls_type2(X_ts,M1,M2,sigma,c1,c2);

    R=R';

    %Calculating the time for testing (T2F)
    Timets = cputime - starttime;

    %Calculating RMSE for the testing set (T2F)
    RMSEts = errperf(D_ts,R,'rmse');
    MAEts = errperf(D_ts,R,'mae');
    MAPEts = errperf(D_ts,R,'mape');
    Er = D_ts-R;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the testing set (T2F)
    varx=sum(R.^2)/n_ts - (sum(R)^2)/(n_ts^2);
    vary=sum(D_ts.^2)/n_ts - (sum(D_ts)^2)/(n_ts^2);
    covxy=sum(R.*D_ts)/n_ts  - (sum(R)*sum(D_ts))/(n_ts^2);
    ccts=covxy/(sqrt(varx * vary));

    T2Fcctr = [T2Fcctr cctr];
    T2Fccts = [T2Fccts ccts];

end

% ============================================================
% Plotting results for training and testing for the ANN
% ============================================================
figure('name','Optimal Value of Alpha for T2Fuzzy','NumberTitle','off')
q=1:length(T2Fcctr);
plot(q,T2Fcctr,'b+-');              %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(q,T2Fccts,'r*-');              %plot(tr_Y(:,s),
xlabel(['Values of Apha'],'FontSize',12);
ylabel(['R-Square'],'FontSize',12);
grid on;
title('Optimal Value of Alpha for T2Fuzzy','FontSize',12)
%title('Optimal Number of Hidden Neurons for ANN (Permeability Data 1)','FontSize',12)
legend('Training R-Square','Testing R-Square')


disp(' ');
disp('----------------END OF STRATIFICATION TEST---------------- ');
```

A.2 Main Program

```
%This single M-file runs 4 AI Techniques: ANN, FN, SVM and Type-2 Fuzzy.
% The purpose is to AUTOMATICALLY determine which stratification option is the best.
% From 99 to 1 with increment of 1.

clear all; close all; clc;
warning off all;
%format short;

disp('------------------------------------------------------------------');
disp('WELCOME to the STRATIFICATION TEST PROGRAM');
disp('IMPLEMENTED ON ANN, FN, SVM and T2FLS');
disp('------------------------------------------------------------------');
disp(' ');
disp('Loading Data ... Press any key to continue');
disp(' ');
%pause;

%Reading Depth
depth                                                                                =
xlsread('/home/ecc_16/harbiwa/Desktop/ANNCode/WaelCodes/input_target_data_selection.xl
sx','a3:a204');

%Reading Data
X                                                                                    =
xlsread('/home/ecc_16/harbiwa/Desktop/ANNCode/WaelCodes/input_target_data_selection.xl
sx','b3:d204');

%Reading Target
D                                                                                    =
xlsread('/home/ecc_16/harbiwa/Desktop/ANNCode/WaelCodes/input_target_data_selection.xl
sx','e3:e204');

data=[X,D];

disp(' ');
disp('Data Successfully Loaded');
disp(' ');

[n,p]=size(X);

fprintf('Number of observations: = %5.0f \n',n),
fprintf('Number of features: = %5.0f \n',p),
disp(' ');

% ANN parameters
numHiddenNeurons = 6; %%This is the optimal (Fatai) %40; %10;  % Adjust as desired
net.trainParam.epochs = 1000;

%SVM parameters
C =  14000; %100; %450; %1000; %1500; %2500; %3500; %4500;
lambda =0.0011; % 1e-7;
epsilon = 0.0007; %0.04
kerneloption = 0.30;
kernel='poly';
verbose=1;

%FunNet parameters
k=1;

%T2F parameters
alpha= 0.048; %0.2; %1.0; % 0.15; %1.0; %0.1; %1.5;   %0.5; %0.01;   %0.0001;%0.001;
%c1;
alpha4=alpha;
```

```
choice = 0.7;

%Stratification Initialization in a Loop
%for i=0.9:0.1:-0.1
% disp('-----------------------------------------------------');
% num_of_runs = input('Enter desired number of runs (5 recommended due to time):> ');
% disp('-----------------------------------------------------');

% disp('-----------------------------------------------------');
% choice = input('Enter desired Stratification (Must be a fraction):> ');
% disp('-----------------------------------------------------');

%for choice=96:-1:92

%Stratification initialization
choice = 0.7;

index=1:n;index=index';

s0=stratif(n,ceil(choice*n));
X_tr = X(s0,:); D_tr = D(s0,:); data_tr=data(s0,:); depth_tr=depth(s0,:);
index0=index; index0(s0)=[];
X_ts  =  X(index0,:);  D_ts  =  D(index0,:);  data_ts  =  data(index0,:);  depth_ts  =
depth(index0,:);

    %Stratification Process finished

    [n_tr,p]=size(X_tr);

    %Transposing the input and output for ANN
    X_tr1=X_tr';
    D_tr1=D_tr';

    % ANN Starts here
    disp(' ');
    disp('ANN Section Started');
    disp(' ');

    tr_starttime = cputime;                % Set time clock

    % Create a FFBP ANN
net = feedforwardnet(numHiddenNeurons);
net.trainParam.epochs = 50;

net = train(net,X_tr1,D_tr1);

view(net)


%Evaluate with Training data
outputtr = net(X_tr1);

Timetr = cputime - tr_starttime;

    % ============================================================
    %Corellation between training output and values predicted with training data
    % ============================================================
    varx=sum(outputtr.^2)/n_tr - (sum(outputtr)^2)/(n_tr^2);
    vary=sum(D_tr1.^2)/n_tr - (sum(D_tr1)^2)/(n_tr^2);
    covxy=sum(outputtr.*D_tr1)/n_tr  - (sum(outputtr)*sum(D_tr1))/(n_tr^2);
    cctr=covxy/(sqrt(varx * vary));
```

```
    % ========================================================
    % Calculating Root mean Square Error for Training
    % ========================================================
    RMSEtr = errperf(D_tr1,outputtr,'rmse');
    MAEtr = errperf(D_tr1,outputtr,'mae');
    MAPEtr = errperf(D_tr1,outputtr,'mape');
    Er = D_tr1-outputtr;
    MIN = min(Er);
    MAX = max(Er);

    %Displaying CC, RMSE and ET for Training
    disp(' ');
    fprintf('Correlation Coefficient for the ANN (Training): %f \n',cctr),
    fprintf('Root Mean Square Errors for the ANN (Training): %5.5f \n',RMSEtr),
    fprintf('Mean Absolute Errors for the ANN (Training): %5.5f \n',MAEtr),
    fprintf('Mean Absolute Percentage Errors for the ANN (Training): %5.5f
\n',MAPEtr),
    fprintf('Minimum Error for the ANN (Training): %5.5f \n',MIN),
    fprintf('Maximum Error for the ANN (Training): %5.5f \n',MAX),
    fprintf('Computation Time for the ANN (Training): %5f \n',Timetr),
    disp(' ');

    %============================================================
    %45 degree Plotting results for training for ANN
    %============================================================
    figure('name','Measured Versus Predicted Training Output using
ANN','NumberTitle','off')
    %q=1:length(D_tr1);
    plot(D_tr1, outputtr,'b*');               %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot of Measured and Predicted SW (Training) using ANN','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % ========================================================
% Plotting results for training for the ANN Model
% ========================================================
figure('name','ANN Model Training','NumberTitle','off')
plot(outputtr,depth_tr,'b-*');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(D_tr1,depth_tr,'r-o');              %plot(tr_Y(:,s),
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Training using ANN Model','FontSize',12)
legend('Measured','Predicted')
% ========================================================

    % ========================================================
% Error Plotting results for training for the ANN Model
% ========================================================
    figure('name','Error for ANN Training','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
    xlabel('Number of data sample');
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope

    %Testing begins here
    [n_ts,p]=size(X_ts);
```

```
    %Also, Transposing the input and output for ANN
    X_ts1=X_ts';
    D_ts1=D_ts';

    ts_starttime = cputime;                % Set time clock

    %Really Testing with Unseen Data
    outputts = net(X_ts1);

    % ========================================================
    % Get computation time for training
    % ========================================================
    Timets = cputime - ts_starttime;

    % ========================================================
    %Corellation between training output and values predicted with Testing data
    % ========================================================
    varx=sum(outputts.^2)/n_ts - (sum(outputts)^2)/(n_ts^2);
    vary=sum(D_ts1.^2)/n_ts - (sum(D_ts1)^2)/(n_ts^2);
    covxy=sum(outputts.*D_ts1)/n_ts  - (sum(outputts)*sum(D_ts1))/(n_ts^2);
    ccts=covxy/(sqrt(varx * vary));

    % ========================================================
    % Calculating Root mean Square Error for Training
    % ========================================================
    RMSEts = errperf(D_ts1,outputts,'rmse');
    MAEts = errperf(D_ts1,outputts,'mae');
    MAPEts = errperf(D_ts1,outputts,'mape');
    Er = D_ts1-outputts;
    MIN = min(Er);
    MAX = max(Er);

    %Displaying CC, RMSE and ET for Testing
    disp(' ');
    fprintf('Correlation Coefficient for the ANN (Testing): %f \n',ccts),
    fprintf('Root Mean Square Errors for the ANN (Testing): %5.5f \n',RMSEts),
    fprintf('Mean Absolute Errors for the ANN (Testing): %5.5f \n',MAEts),
    fprintf('Mean Absolute Percentage Errors for the ANN (Testing): %5.5f \n',MAPEts),
    fprintf('Minimum Error for the ANN (Testing): %5.5f \n',MIN),
    fprintf('Maximum Error for the ANN (Testing): %5.5f \n',MAX),
    fprintf('Computation Time for the ANN (Testing): %5f \n',Timets),
    disp(' ');

    %=========================================================
    %45 degree Plotting results for testing for ANN
    %=========================================================
    figure('name','Measured     Versus     Predicted     Testing     Output     using
ANN','NumberTitle','off')
    %q=1:length(D_ts1);
    plot(D_ts1, outputts,'b*');               %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot of Measured and Predicted SW (Testing) using ANN','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % ========================================================
% Plotting results for testing for the ANN Model
% ========================================================
figure('name','ANN Model Testing','NumberTitle','off')
plot(outputts,depth_ts,'b-*');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(D_ts1,depth_ts,'r-o');              %plot(tr_Y(:,s),
```

```
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Testing using ANN Model','FontSize',12)
legend('Measured','Predicted')
% ========================================================

    % ========================================================
% Error Plotting results for testing for the ANN Model
% ========================================================
    figure('name','Error for ANN Testing','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
    xlabel('Number of data sample');
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope


    %FunNet Starts Here
    disp(' ');
    disp('FunNet Started');
    disp(' ');

    starttime = cputime;

    % Training Functional Networks
    [Best_Model, MDL, RMSE, Best_Sub] = fn_bf(D_tr(:,k),X_tr);
    FunNets_Coeff = X_tr(:, Best_Sub)\D_tr(:,k);

    %Calculating the time for training (FunNet)
    Timetr = cputime - starttime;

    % Testing the Training
    y0 = X_tr(:, Best_Sub)*FunNets_Coeff;

    %Calculating RMSE for the training set (FunNet)
    RMSEtr = errperf(D_tr,y0,'rmse');
    MAEtr = errperf(D_tr,y0,'mae');
    MAPEtr = errperf(D_tr,y0,'mape');
    Er = D_tr-y0;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the training set (FunNet)
    varx=sum(y0.^2)/n_tr - (sum(y0)^2)/(n_tr^2);
    vary=sum(D_tr.^2)/n_tr - (sum(D_tr)^2)/(n_tr^2);
    covxy=sum(y0.*D_tr)/n_tr  - (sum(y0)*sum(D_tr))/(n_tr^2);
    cctr=covxy/(sqrt(varx * vary));

    %Displaying CC, RMSE and ET for Training
    disp(' ');
    fprintf('Correlation Coefficient for the FN (Training): %f \n',cctr),
    fprintf('Root Mean Square Errors for the FN (Training): %5.5f \n',RMSEtr),
    fprintf('Mean Absolute Errors for the FN (Training): %5.5f \n',MAEtr),
    fprintf('Mean Absolute Percentage Errors for the FN (Training): %5.5f \n',MAPEtr),
    fprintf('Minimum Error for the FN (Training): %5.5f \n',MIN),
    fprintf('Maximum Error for the FN (Training): %5.5f \n',MAX),
    fprintf('Computation Time for the FN (Training): %5f \n',Timetr),
    disp(' ');

    %========================================================
    %45 degree Plotting results for training for FN
```

```
    %============================================================
    figure('name','Measured     Versus     Predicted     Training     Output     using
FN','NumberTitle','off')
    %q=1:length(D_tr1);
    plot(D_tr, y0,'b*');               %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot of Measured and Predicted SW (Training) using FN','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % ============================================================
% Plotting results for training for the FN Model
% ============================================================
figure('name','FN Model Training','NumberTitle','off')
plot(y0,depth_tr,'b-*');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(D_tr,depth_tr,'r-o');               %plot(tr_Y(:,s),
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Training using FN Model','FontSize',12)
legend('Measured','Predicted')
% ============================================================

    % ============================================================
% Error Plotting results for training for the FN Model
% ============================================================
    figure('name','Error for FN Training','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
    xlabel('Number of data sample');
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope

    starttime = cputime;

    % Testing with unseen Data
    ys = X_ts(:, Best_Sub)*FunNets_Coeff;

    %Cummulating the time for testing (FunNet)
    Timets = cputime - starttime;

    %Calculating RMSE for the testing set (FunNet)
    RMSEts = errperf(D_ts,ys,'rmse');
    MAEts = errperf(D_ts,ys,'mae');
    MAPEts = errperf(D_ts,ys,'mape');
    Er = D_ts-ys;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the testing set (FunNet)
    varx=sum(ys.^2)/n_ts - (sum(ys)^2)/(n_ts^2);
    vary=sum(D_ts.^2)/n_ts - (sum(D_ts)^2)/(n_ts^2);
    covxy=sum(ys.*D_ts)/n_ts  - (sum(ys)*sum(D_ts))/(n_ts^2);
    ccts=covxy/(sqrt(varx * vary));


    %Displaying CC, RMSE and ET for Testing
    disp(' ');
    fprintf('Correlation Coefficient for the FN (Testing): %f \n',ccts),
    fprintf('Root Mean Square Errors for the FN (Testing): %5.5f \n',RMSEts),
```

```
    fprintf('Mean Absolute Errors for the FN (Testing): %5.5f \n',MAEts),
    fprintf('Mean Absolute Percentage Errors for the FN (Testing): %5.5f \n',MAPEts),
    fprintf('Minimum Error for the FN (Testing): %5.5f \n',MIN),
    fprintf('Maximum Error for the FN (Testing): %5.5f \n',MAX),
    fprintf('Computation Time for the FN (Testing): %5f \n',Timets),
    disp(' ');

    %=========================================================
    %45 degree Plotting results for testing for FN
    %=========================================================
    figure('name','Measured     Versus     Predicted     Testing     Output     using
FN','NumberTitle','off')
    %q=1:length(D_ts1);
    plot(D_ts, ys,'b*');               %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot of Measured and Predicted SW (Testing) using FN','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % =========================================================
% Plotting results for testing for the FN Model
% =========================================================
figure('name','FN Model Testing','NumberTitle','off')
plot(ys,depth_ts,'b-*');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(D_ts,depth_ts,'r-o');               %plot(tr_Y(:,s),
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Testing using FN Model','FontSize',12)
legend('Measured','Predicted')
% =========================================================

    % =========================================================
% Error Plotting results for testing for the FN Model
% =========================================================
    figure('name','Error for FN Testing','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
    xlabel('Number of data sample');
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope


    % SVM Starts Here
    disp(' ');
    disp('SVM Started');
    disp(' ');

    % Identifying the input and output for training
    x=X_tr;
    ytr=D_tr;

    % Initializing the CPU Start Time
    starttime = cputime;

    %Training SVM
    [xsup,ysup,w,w0] = svmreg(x,ytr,C,epsilon,kernel,kerneloption,lambda,verbose);

    %Calculating the time for training (SVM)
    Timetr = cputime - starttime;
```

```
    % Testing the Training
    ypredtr = svmval(x,xsup,w,w0,kernel,kerneloption);

    %Calculating RMSE for the training set (SVM)
    RMSEtr = errperf(ytr,ypredtr,'rmse');
    MAEtr = errperf(ytr,ypredtr,'mae');
    MAPEtr = errperf(ytr,ypredtr,'mape');
    Er = ytr-ypredtr;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the training set (SVM)
    varx=sum(ypredtr.^2)/n_tr - (sum(ypredtr)^2)/(n_tr^2);
    vary=sum(ytr.^2)/n_tr - (sum(ytr)^2)/(n_tr^2);
    covxy=sum(ypredtr.*ytr)/n_tr  - (sum(ypredtr)*sum(ytr))/(n_tr^2);
    cctr=covxy/(sqrt(varx * vary));

    %Displaying CC, RMSE and ET for Training
    disp(' ');
    fprintf('Correlation Coefficient for the SVM (Training): %f \n',cctr),
    fprintf('Root Mean Square Errors for the SVM (Training): %5.5f \n',RMSEtr),
    fprintf('Mean Absolute Errors for the SVM (Training): %5.5f \n',MAEtr),
    fprintf('Mean  Absolute  Percentage  Errors  for  the  SVM  (Training):  %5.5f
\n',MAPEtr),
    fprintf('Minimum Error for the SVM (Training): %5.5f \n',MIN),
    fprintf('Maximum Error for the SVM (Training): %5.5f \n',MAX),
    fprintf('Computation Time for the SVM (Training): %5f \n',Timetr),
    disp(' ');

    %============================================================
    %45 degree Plotting results for training for SVM
    %============================================================
    figure('name','Measured    Versus    Predicted    Training    Output    using
SVM','NumberTitle','off')
    %q=1:length(D_tr1);
    plot(ytr, ypredtr,'b*');              %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot of Measured and Predicted SW (Training) using SVM','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % ============================================================
% Plotting results for training for the SVM Model
% ============================================================
figure('name','SVM Model Training','NumberTitle','off')
plot(ypredtr,depth_tr,'b-*');              %plot(tr_Y(:,s),
hold on;            %plot(tr_Y(:,s),
plot(ytr,depth_tr,'r-o');             %plot(tr_Y(:,s),
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Training using SVM Model','FontSize',12)
legend('Measured','Predicted')
% ============================================================

    % ============================================================
% Error Plotting results for training for the SVM Model
% ============================================================
    figure('name','Error for SVM Training','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
```

```
    xlabel('Number of data sample');
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope


    % Identifying the test data
    x=X_ts;
    yts=D_ts;

    % Initializing the CPU Start Time
    starttime = cputime;

    % Testing with unseen Data
    ypredts = svmval(x,xsup,w,w0,kernel,kerneloption);

    %Calculating the time for testing (SVM)
    Timets = cputime - starttime;

    %Calculating RMSE for the testing set (SVM)
    RMSEts = errperf(yts,ypredts,'rmse');
    MAEts = errperf(yts,ypredts,'mae');
    MAPEts = errperf(yts,ypredts,'mape');
    Er = yts-ypredts;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the testing set (SVM)
    varx=sum(ypredts.^2)/n_ts - (sum(ypredts)^2)/(n_ts^2);
    vary=sum(yts.^2)/n_ts - (sum(yts)^2)/(n_ts^2);
    covxy=sum(ypredts.*yts)/n_ts  - (sum(ypredts)*sum(yts))/(n_ts^2);
    ccts=covxy/(sqrt(varx * vary));

    %Displaying CC, RMSE and ET for Testing
    disp(' ');
    fprintf('Correlation Coefficient for the SVM (Testing): %f \n',ccts),
    fprintf('Root Mean Square Errors for the SVM (Testing): %5.5f \n',RMSEts),
    fprintf('Mean Absolute Errors for the SVM (Testing): %5.5f \n',MAEts),
    fprintf('Mean Absolute Percentage Errors for the SVM (Testing): %5.5f \n',MAPEts),
    fprintf('Minimum Error for the SVM (Testing): %5.5f \n',MIN),
    fprintf('Maximum Error for the SVM (Testing): %5.5f \n',MAX),
    fprintf('Computation Time for the SVM (Testing): %5f \n',Timets),
    disp(' ');

    %============================================================
    %45 degree Plotting results for testing for SVM
    %============================================================
    figure('name','Measured     Versus     Predicted     Testing     Output     using
SVM','NumberTitle','off')
    %q=1:length(D_ts1);
    plot(yts, ypredts,'b*');               %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot of Measured and Predicted SW (Testing) using SVM','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % ============================================================
% Plotting results for testing for the SVM Model
% ============================================================
figure('name','SVM Model Testing','NumberTitle','off')
plot(yts,depth_ts,'b-*');              %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
```

```
plot(ypredts,depth_ts,'r-o');               %plot(tr_Y(:,s),
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Testing using SVM Model','FontSize',12)
legend('Measured','Predicted')
% ============================================================

    % ============================================================
% Error Plotting results for testing for the SVM Model
% ============================================================
    figure('name','Error for SVM Testing','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
    xlabel('Number of data sample');
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope

    % Type-2 Fuzzy Starts Here
    disp(' ');
    disp('T2-Fuzzy Started');
    disp(' ');

    % Initializing Variables and Constants
    M1=X_tr;M2=X_tr;c1=D_tr;
    c2=c1;

    for f=1:size(X_tr,1)
        for j=1:size(X_tr,2)
            sigma(f,j)=(1.4272 -M1(f,j))/2; % This is option [3]
        end
    end

    sn1 = std(X_tr);
    sn2=sn1;

    % Initializing CPU Start Time
    starttime = cputime;

    %Training T2F

[M1,M2,c1,c2,sigma,sn1,sn2]=train_nsfls2(X_tr,D_tr,M1,M2,sigma,c1,c2,sn1,sn2,alpha,alp
ha4); %train_nsfls2

    %Cummulating the time for training (T2F)
    Timetr = cputime - starttime;

    % Testing the Training
    [R1,R2,R]=sfls_type2(X_tr,M1,M2,sigma,c1,c2);

    R=R';

    %Calculating RMSE for the training set (T2F)
    RMSEtr = errperf(D_tr,R,'rmse');
    MAEtr = errperf(D_tr,R,'mae');
    MAPEtr = errperf(D_tr,R,'mape');
    Er = D_tr-R;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the training set (T2F)
    varx=sum(R.^2)/n_tr - (sum(R)^2)/(n_tr^2);
```

```
    vary=sum(D_tr.^2)/n_tr - (sum(D_tr)^2)/(n_tr^2);
    covxy=sum(R.*D_tr)/n_tr  - (sum(R)*sum(D_tr))/(n_tr^2);
    cctr=covxy/(sqrt(varx * vary));

    %Displaying CC, RMSE and ET for Training
    disp(' ');
    fprintf('Correlation Coefficient for the T2F (Training): %f \n',cctr),
    fprintf('Root Mean Square Errors for the T2F (Training): %5.5f \n',RMSEtr),
    fprintf('Mean Absolute Errors for the T2F (Training): %5.5f \n',MAEtr),
    fprintf('Mean   Absolute   Percentage   Errors   for   the   T2F   (Training):   %5.5f
\n',MAPEtr),
    fprintf('Minimum Error for the T2F (Training): %5.5f \n',MIN),
    fprintf('Maximum Error for the T2F (Training): %5.5f \n',MAX),
    fprintf('Computation Time for the T2F (Training): %5f \n',Timetr),
    disp(' ');

    %============================================================
    %45 degree Plotting results for training for SVM
    %============================================================
    figure('name','Measured     Versus     Predicted     Training     Output     using
T2FLS','NumberTitle','off')
    %q=1:length(D_tr1);
    plot(D_tr, R,'b*');               %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot    of    Measured    and    Predicted    SW    (Training)    using
T2FLS','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % ============================================================
% Plotting results for training for the T2FLS Model
% ============================================================
figure('name','T2FLS Model Training','NumberTitle','off')
plot(R,depth_tr,'b-*');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(D_tr,depth_tr,'r-o');               %plot(tr_Y(:,s),
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Training using T2FLS Model','FontSize',12)
legend('Measured','Predicted')
% ============================================================

    % ============================================================
% Error Plotting results for training for the T2FLS Model
% ============================================================
    figure('name','Error for T2FLS Training','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
    xlabel('Number of data sample');
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope

    % Initializing CPU Start Time
    starttime = cputime;

    % Testing with unseen Data
    [R1,R2,R]=sfls_type2(X_ts,M1,M2,sigma,c1,c2);

    R=R';
```

```
    %Calculating the time for testing (T2F)
    Timets = cputime - starttime;

    %Calculating RMSE for the testing set (T2F)
    RMSEts = errperf(D_ts,R,'rmse');
    MAEts = errperf(D_ts,R,'mae');
    MAPEts = errperf(D_ts,R,'mape');
    Er = D_ts-R;
    MIN = min(Er);
    MAX = max(Er);

    %Calculating Correlation Coeficient for the testing set (T2F)
    varx=sum(R.^2)/n_ts - (sum(R)^2)/(n_ts^2);
    vary=sum(D_ts.^2)/n_ts - (sum(D_ts)^2)/(n_ts^2);
    covxy=sum(R.*D_ts)/n_ts  - (sum(R)*sum(D_ts))/(n_ts^2);
    ccts=covxy/(sqrt(varx * vary));

    %Displaying CC, RMSE and ET for Testing
    disp(' ');
    fprintf('Correlation Coefficient for the T2F (Testing): %f \n',ccts),
    fprintf('Root Mean Square Errors for the T2F (Testing): %5.5f \n',RMSEts),
    fprintf('Mean Absolute Errors for the T2F (Testing): %5.5f \n',MAEts),
    fprintf('Mean Absolute Percentage Errors for the T2F (Testing): %5.5f \n',MAPEts),
    fprintf('Minimum Error for the T2F (Testing): %5.5f \n',MIN),
    fprintf('Maximum Error for the T2F (Testing): %5.5f \n',MAX),
    fprintf('Computation Time for the T2F (Testing): %5f \n',Timets),
    disp(' ');

    %============================================================
    %45 degree Plotting results for testing for T2FLS
    %============================================================
    figure('name','Measured     Versus     Predicted     Testing     Output     using
T2FLS','NumberTitle','off')
    %q=1:length(D_ts1);
    plot(D_ts, R,'b*');               %plot(tr_Y(:,s),
    xlabel(['Measured SW'],'FontSize',12);
    ylabel(['Predicted SW'],'FontSize',12);
    title('Crossplot    of    Measured    and    Predicted    SW    (Testing)    using
T2FLS','FontSize',12)
    %legend('Measured','Predicted')
    refline(1,0);% Addding Reference Line with 0.0 degree slope

    % ============================================================
% Plotting results for testing for the T2FLS Model
% ============================================================
figure('name','T2FLS Model Testing','NumberTitle','off')
plot(D_ts,depth_ts,'b-*');               %plot(tr_Y(:,s),
hold on;             %plot(tr_Y(:,s),
plot(R,depth_ts,'r-o');               %plot(tr_Y(:,s),
xlabel(['SW'],'FontSize',12);
ylabel(['Depth(ft)'],'FontSize',12);
grid on;
title('SW Testing using T2FLS Model','FontSize',12)
legend('Measured','Predicted')
% ============================================================

    % ============================================================
% Error Plotting results for testing for the T2FLS Model
% ============================================================
    figure('name','Error for T2FLS Testing','NumberTitle','off')
    plot(Er,'ro')
    title ('Network Model Error Distribution');
    xlabel('Number of data sample');
```

```
    ylabel('Errors');
    grid on
    refline(0,0);% Addding Reference Line with 0.0 degree slope

    disp(' ');
    disp('-----------------------------------------------------------------');
    disp(' ');
%end
disp(' ');
disp('----------------END OF STRATIFICATION TEST----------------- ');
```

6.5 Vitae

Professional Experience

Saudi ARAMCO 2004 – Present

PE Systems Analyst

First Line Support - Provide various technical support services for the company vendor's applications such as OFM (2004-2007), Geolog and Emeraude (2007-2013), including problem analysis and resolution for over +300 professionals.

Technical Work:

- Responsible on system administration, maintenance, customization and enforcing company standards
- Trouble shooting and resolved technical operational problems of the application
- Provide continuous technical support for applications to ensure optimal stability, usability and dependability

Technical Skills

- IT Project manager for several in-house development projects
- Comprehensive understanding of the software development lifecycle (SDLC)
- Application training

Technical Support:

- Responsible for technical support for users
- Supervise a professional support staff of three
- Develop training modules and documentation for users
- Conduct training courses for users
- Highly skilled in resolving technical customer issues in a Help Desk or Technical Support department
- Work with programming staff in designing new functionality and enhancement in the software.
- utilization of BMC remedy action ticketing system for tracking all users requests
- Advance understanding of designing Oracle database model for application

- Retained close partnership with software vendors to ensure timely support whenever necessary.